

ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

GIOVANI PARENTE FARIAS

PLAN RECOGNITION AND FAILURE PREDICTION FOR
AMBIENT ASSISTED LIVING

Porto Alegre
2017

PÓS-GRADUAÇÃO - *STRICTO SENSU*



Pontifícia Universidade Católica
do Rio Grande do Sul

**PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
FACULTY OF INFORMATICS
COMPUTER SCIENCE GRADUATE PROGRAM**

**PLAN RECOGNITION AND
FAILURE PREDICTION FOR
AMBIENT ASSISTED LIVING**

GIOVANI PARENTE FARIAS

Dissertation submitted to the Pontifical
Catholic University of Rio Grande
do Sul in partial fulfillment of the
requirements for the degree of Ph. D.
in Computer Science.

Advisor: Prof. Rafael Heitor Bordini
Co-Advisor: Prof. Felipe Rech Meneguzzi

**Porto Alegre
2017**

Ficha Catalográfica

F224p Farias, Giovani Parente

Plan recognition and failure prediction for ambient assisted living / Giovani Parente Farias . – 2017.

184 f.

Tese (Doutorado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Rafael Heitor Bordini.

Co-orientador: Prof. Dr. Felipe Rech Meneguzzi.

1. Activity Recognition. 2. Plan Recognition. 3. Multiple-goal Recognition. 4. Failure Prediction. I. Bordini, Rafael Heitor. II. Meneguzzi, Felipe Rech. III. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).

Bibliotecários responsáveis: Marcelo Votto Teixeira CRB-10/1974 e Michelângelo Viana CRB-10/1306

Giovani Parente Farias

**Plan Recognition and Failure Prediction for
Ambient Assisted Living**

This Thesis has been submitted in partial fulfillment of the requirements for the degree of Doctor of Computer Science, of the Graduate Program in Computer Science, School of Computer Science of the Pontifícia Universidade Católica do Rio Grande do Sul.

Sanctioned on August 31, 2017.

COMMITTEE MEMBERS:

Prof. Dr. Avelino Francisco Zorzo (PPGCC/PUCRS)

Profa. Dra. Diana Francisca Adamatti (PPGMC/FURG)

Prof. Dr. Luís Alvaro de Lima Silva (PPGI/UFSM)

Prof. Dr. Felipe Rech Meneguzzi (PPGCC/PUCRS - Co-Advisor)

Prof. Dr. Rafael Heitor Bordini (PPGCC/PUCRS - Advisor)

“The wisest mind has something yet to learn.”
(George Santayana)

ACKNOWLEDGMENTS

Apart from the efforts of myself, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been fundamental in the successful completion of this thesis. Firstly, I would like to express my sincere gratitude to my advisor for the continuous support of my PhD study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and provided me with the tools to choose the right direction and successfully complete my thesis.

It has been a period of intense learning for me, not only in the scientific area but also on a personal level. I would like to thank my friends for their wonderful collaboration and for all the fun times we spent together. They have been my family away from home and my life has been so much richer thanks to them.

Finally, I must express my very profound gratitude to my parents for their continued support and encouragement. They have devoted their life to my sister and me and highly prioritised our education. They have given me all the opportunities in the world to choose what I want to be and supported me all the way. Words cannot express how grateful I am to my parents for all of the sacrifices that they have made on my behalf. This accomplishment would not have been possible without them.

RECONHECIMENTO DE PLANOS E PREDIÇÃO DE FALHAS EM AMBIENTE ASSISTIDO

RESUMO

O processo de inferir os planos/objetivos de um agente com base na observação de suas ações é conhecido como reconhecimento de plano. Prever as intenções humanas é um dos objetivos atuais da Inteligência Artificial; o reconhecimento de plano contribui para esse objetivo ao analisar como as observações em baixo nível sobre agentes e meio ambiente podem ser associadas a uma descrição em alto nível do plano. A maioria das abordagens para reconhecimento de plano, em ambientes reais, baseiam-se em regras construídas manualmente, onde a base de conhecimento é representada como uma biblioteca de planos para reconhecimento de atividades e planos. Além disso, essas abordagens geralmente não têm capacidade de incorporar dependências temporais complexas, assumindo a hipótese irrealista de que um agente executa apenas uma atividade por vez e que a sequência de ações é executada de forma coerente para alcançar um único objetivo. Além disso, o conhecimento incompleto sobre o comportamento do agente e a similaridade entre a execução de vários planos geram múltiplas hipóteses sobre o(s) plano(s) que são consistentes com as observações. Este trabalho aborda os problemas para reconhecer múltiplos planos em ambientes reais, aprender a duração de uma atividade e detectar anomalias na execução de um plano. Tratamos o problema de desambiguar múltiplas hipóteses e detectamos anomalias na sequência de execução do plano explorando tanto a organização hierárquica inerente das atividades quanto horário e duração esperados, desenvolvendo um algoritmo eficiente para filtrar hipóteses aplicando restrições temporais e no comprimento do caminho. Apresentamos uma série de experimentos mostrando que, além de abordar limitações dos algoritmos de reconhecimento de planos tradicionais, nossa abordagem de filtrar hipóteses pode melhorar significativamente a precisão do algoritmo de reconhecimento. Os experimentos incluem bibliotecas de planos geradas sinteticamente, bem como bibliotecas e observações obtidas a partir de conjuntos de dados do mundo real, útil no contexto de ambiente assistido.

Palavras-Chave: reconhecimento de atividade, reconhecimento de plano, reconhecimento de múltiplos objetivos, predição de falha.

PLAN RECOGNITION AND FAILURE PREDICTION FOR AMBIENT ASSISTED LIVING

ABSTRACT

The process of inferring agent's plans/goals from their observed actions is known as plan recognition. Predicting human intentions is one of the ultimate goals of Artificial Intelligence; plan recognition contributes to this goal by analysing how low-level observations about agents and environment can be associated with a high-level plan description. Most approaches to plan recognition, in realistic environments, are based on manually constructed rules, where the knowledge base is represented as a plan library for recognising activities and plans. Besides, these approaches do not usually have the ability to incorporate complex temporal dependencies, and they take the unrealistic assumption that an agent carries out only one activity at a time and the sequence of actions is all coherently executed towards a single goal. Moreover, the incomplete knowledge about the agent's behaviour and the similarity among several plan execution generate multiple hypotheses about the agent's plan(s) that are consistent with the observations. This work addresses the problems of recognising multiple plans in realistic environments, learning activity duration, and detecting anomalies in plan execution. We deal with problems related to disambiguation of multiple hypotheses and detecting anomalies in plan sequence by exploiting both the inherent hierarchical organisation of activities and their expected time and duration, developing an efficient algorithm to filter the hypotheses by applying temporal and path length constraints. We present a number of experimental results showing that, besides addressing those limitations of traditional plan recognition algorithms, our filtering approach can significantly improve the accuracy of the underlying plan recognition algorithm. The experiments include a number of synthetically generated plan libraries as well as plan libraries and observations obtained from a real-world dataset useful in the context of ambient assisted living.

Keywords: activity recognition, plan recognition, multiple-goal recognition, failure prediction.

LIST OF FIGURES

Figure 2.1 – Plan library example of robocup soccer teams [AZK05].	36
Figure 2.2 – Example of a FDT associated with a plan library [Avr09].	38
Figure 3.1 – Integration of failure predictor components.	45
Figure 3.2 – Example of a plan library based on ADL.	50
Figure 4.1 – Example of a plan library with $g=2$, $d=3$, $b=1$ and $B=3$	54
Figure 4.2 – Part of a plan library with features valued $f=1$, $c=2$ and $F=3$. . .	55
Figure 4.3 – Examples of branch creation, solid lines represent decomposition links and dashed lines represent sequential links.	56
Figure 4.4 – Example of duplication in a plan library where top-level plan p_n is a duplication of p_1 differing only in the leaf nodes.	56
Figure 4.5 – Example of a plan library tree created by the <i>Plan Library Genera-</i> <i>tor</i> with $g=2$, $d=3$, $b=1$, $B=3$, $\Phi=0.5$, $F=3$, $f=1$, $c=2$, and $\Psi=0$	59
Figure 4.6 – Average matching runtime of SBR varying d and Φ	63
Figure 4.7 – FDT training time varying d and Φ	63
Figure 4.8 – Average matching runtime of SBR varying g and f	64
Figure 4.9 – FDT training time varying g and f	65
Figure 4.10 – Average matching runtime of SBR varying B and f	65
Figure 4.11 – FDT training time varying B and f	66
Figure 4.12 – Average matching runtime of SBR varying f and c	66
Figure 4.13 – FDT training time varying f and c	67
Figure 5.1 – Temporal multiple-plan recognition architecture.	70
Figure 5.2 – Example of node types and their positions in a plan library.	72
Figure 6.1 – Example of sequential plans.	94
Figure 6.2 – Example of parallel plans.	94
Figure 6.3 – Example of interleaved plans.	95
Figure 6.4 – Example of a recognition process with various recognition episodes. .	96
Figure 6.6 – Boxplot graphs comparing SBR and PEC precision performance switching path length (δ) for subject one using partial dataset.	102
Figure 6.8 – Boxplot graphs comparing SBR and PEC precision performance switching path length (δ) for subject two using complete dataset.	105
Figure A.2 – PR space graphs switching parallel plans (\mathcal{P}) and path length (δ) for subject one using complete dataset where all plans have defined hour.	130

Figure A.4 – PR space graphs switching parallel plans (\mathcal{P}) and path length (δ) for subject one using partial dataset where all plans have defined hour. . .	131
Figure A.6 – ROC space graphs switching parallel plans (\mathcal{P}) and path length (δ) for subject one using complete dataset where all plans have defined hour.	132
Figure A.8 – ROC space graphs switching parallel plans (\mathcal{P}) and path length (δ) for subject one using partial dataset where all plans have defined hour. . .	133
Figure B.2 – PR space graphs switching parallel plans (\mathcal{P}) and path length (δ) for subject two using complete dataset where all plans have defined hour.	140
Figure B.4 – PR space graphs switching parallel plans (\mathcal{P}) and path length (δ) for subject two using partial dataset where all plans have defined hour. . .	141
Figure B.6 – ROC space graphs switching parallel plans (\mathcal{P}) and path length (δ) for subject two using complete dataset where all plans have defined hour.	142
Figure B.8 – ROC space graphs switching parallel plans (\mathcal{P}) and path length (δ) for subject two using partial dataset where all plans have defined hour. . .	143

LIST OF TABLES

Table 4.1 – Parameters available to generate the plan library	53
Table 5.1 – Example of temporary table (Υ)	71
Table 5.2 – Example of database (Δ)	73
Table 5.3 – Complete table of number of examples per activity generated by in- direct observation [Mun03]	81
Table 5.4 – Partial table of number of examples per activity generated by indirect observation [Mun03]	81
Table 5.5 – Sample of the sensors.csv file contents	82
Table 5.6 – Sample of the activities.csv file contents	82
Table 5.7 – Data organisation in activities_data.csv	82
Table 5.8 – Part of the data in activities_data.csv file	83
Table 5.9 – Parameters available to generate the temporal plan library	84
Table 6.1 – Confusion matrix	90
Table 6.2 – Example of data organisation and content in activities_data.csv . . .	97
Table 6.3 – PEC, subject one, partial dataset, most plans with undefined hour . .	100
Table 6.4 – SBR, subject one, partial dataset, most plans with undefined hour . .	100
Table 6.5 – PEC, subject one, partial dataset, all plans with defined hour	101
Table 6.6 – SBR, subject one, partial dataset, all plans with defined hour	101
Table 6.7 – PEC, subject two, complete dataset, most plans with undefined hour	103
Table 6.8 – SBR, subject two, complete dataset, most plans with undefined hour	103
Table 6.9 – PEC, subject two, complete dataset, all plans with defined hour . . .	104
Table 6.10 – SBR, subject two, complete dataset, all plans with defined hour . . .	104
Table 6.11 – PEC results for generated simulations based on following parame- ters: $\mathcal{P}=1$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	107
Table 6.12 – SBR results for generated simulations based on following parame- ters: $\mathcal{P}=1$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	107
Table 6.13 – PEC results for generated simulations based on following parame- ters: $\mathcal{P}=3$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	108
Table 6.14 – SBR results for generated simulations based on following parame- ters: $\mathcal{P}=3$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	109
Table 6.15 – PEC results for generated simulations based on following parame- ters: $\mathcal{P}=3$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	110

Table 6.16 – SBR results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	110
Table 7.1 – Characteristics of the main related work	114
Table A.1 – PEC, subject one, complete dataset, most plans with undefined hour	129
Table A.2 – SBR, subject one, complete dataset, most plans with undefined hour	129
Table A.3 – PEC, subject one, complete dataset, all plans with defined hour . . .	129
Table A.4 – SBR, subject one, complete dataset, all plans with defined hour . . .	129
Table A.5 – PEC results for subject one using complete dataset where most plans have undefined hour	134
Table A.6 – SBR results for subject one using complete dataset where most plans have undefined hour	134
Table A.7 – PEC results for subject one using partial dataset where most plans have undefined hour	135
Table A.8 – SBR results for subject one using partial dataset where most plans have undefined hour	135
Table A.9 – PEC results for subject one using complete dataset where all plans have a defined hour	136
Table A.10 – SBR results for subject one using complete dataset where all plans have a defined hour	136
Table A.11 – PEC results for subject one using partial dataset where all plans have a defined hour	137
Table A.12 – SBR results for subject one using partial dataset where all plans have a defined hour	137
Table B.1 – PEC, subject two, partial dataset, most plans with undefined hour . .	139
Table B.2 – SBR, subject two, partial dataset, most plans with undefined hour . .	139
Table B.3 – PEC, subject two, partial dataset, all plans with defined hour	139
Table B.4 – SBR, subject two, partial dataset, all plans with defined hour	139
Table B.5 – PEC results for subject two using complete dataset where most plans have undefined hour	144
Table B.6 – SBR results for subject two using complete dataset where most plans have undefined hour	144
Table B.7 – PEC results for subject two using partial dataset where most plans have undefined hour	145
Table B.8 – SBR results for subject two using partial dataset where most plans have undefined hour	145

Table B.9 – PEC results for subject two using complete dataset where all plans have a defined hour	146
Table B.10 – SBR results for subject two using complete dataset where all plans have a defined hour	146
Table B.11 – PEC results for subject two using partial dataset where all plans have a defined hour	147
Table B.12 – SBR results for subject two using partial dataset where all plans have a defined hour	147
Table C.1 – PEC results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	149
Table C.2 – PEC results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	150
Table C.3 – PEC results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	151
Table C.4 – SBR results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	152
Table C.5 – SBR results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	153
Table C.6 – SBR results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	154
Table C.7 – PEC results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	155
Table C.8 – PEC results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	156
Table C.9 – PEC results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	157
Table C.10 – SBR results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	158
Table C.11 – SBR results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	159
Table C.12 – SBR results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	160
Table C.13 – PEC results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	161
Table C.14 – PEC results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	162

Table C.15 – PEC results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	163
Table C.16 – SBR results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	164
Table C.17 – SBR results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	165
Table C.18 – SBR results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	166
Table C.19 – PEC all results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	167
Table C.20 – PEC all results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	168
Table C.21 – PEC all results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	169
Table C.22 – SBR all results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	170
Table C.23 – SBR all results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	171
Table C.24 – SBR all results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	172
Table C.25 – PEC all results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	173
Table C.26 – PEC all results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	174
Table C.27 – PEC all results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	175
Table C.28 – SBR all results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	176
Table C.29 – SBR all results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	177
Table C.30 – SBR all results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	178
Table C.31 – PEC all results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	179
Table C.32 – PEC all results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$	180

Table C.33 – PEC all results for generated simulations based on following parameters: $\mathcal{P}=3$, $d=11$, $b=1$, $B=2$, $F=10$, $\Phi=1$, $\Psi=0$, $t=1$ and $T=100$	181
Table C.34 – SBR all results for generated simulations based on following parameters: $\mathcal{P}=1$, $d=11$, $b=1$, $B=2$, $F=10$, $\Phi=1$, $\Psi=0$, $t=1$ and $T=100$	182
Table C.35 – SBR all results for generated simulations based on following parameters: $\mathcal{P}=2$, $d=11$, $b=1$, $B=2$, $F=10$, $\Phi=1$, $\Psi=0$, $t=1$ and $T=100$	183
Table C.36 – SBR all results for generated simulations based on following parameters: $\mathcal{P}=3$, $d=11$, $b=1$, $B=2$, $F=10$, $\Phi=1$, $\Psi=0$, $t=1$ and $T=100$	184

LIST OF ACRONYMS

AAL – Ambient Assisted Living

ACC – Accuracy

ADL – Activities of Daily Living

AI – Artificial Intelligence

CRADLE – Cumulative Recognition of Activities and Decreasing Load of Explanations

ERR – Error Rate

ERT – Expected Recognition Time

ESM – Experience Sampling Method

FDR – False Discovery Rate

FDT – Feature Decision Tree

FN – False Negatives

FNR – False Negative Rate

FOR – False Omission Rate

FP – False Positives

FPR – False Positive Rate

GPS – Global Positioning System

HMM – Hidden Markov Model

HSMM – Hidden Semi-Markov Model

ICT – Information and Communication Technologies

IOT – Internet of Things

MAD – Mean Absolute Deviation

MAS – Multi-Agent Systems

MCC – Matthews Correlation Coefficient

MDP – Markov Decision Processes

NPV – Negative Predictive Value

PDDL – Planning Domain Definition Language

PEC – Plan Execution Controller

PHATT – Probabilistic Hostile Agent Task Tracker

PPV – Positive Predictive Value

PR – Precision-Recall

PSC – Plan Selection Counter

RFID – Radio-Frequency IDentification

ROC – Receiver Operating Characteristics

SBR – Symbolic Behaviour Recognition

SHSMM – Switching Hidden Semi-Markov Models

SPRP – Sequential Plan Recognition Process

TN – True Negatives

TNR – True Negative Rate

TP – True Positives

TPR – True Positive Rate

WI-FI – Wireless Fidelity

CONTENTS

1	INTRODUCTION	27
1.1	MOTIVATION	28
1.2	OBJECTIVES	29
1.3	THESIS ORGANISATION	30
2	BACKGROUND	31
2.1	ACTIVITY RECOGNITION	31
2.2	PLAN RECOGNITION	32
2.3	KNOWLEDGE BASE	34
2.4	SYMBOLIC PLAN RECOGNITION	35
2.4.1	PLAN LIBRARY	36
2.4.2	FEATURE DECISION TREE	37
2.5	MULTI-AGENT PLAN RECOGNITION	37
2.6	AMBIENT ASSISTED LIVING	39
3	FAILURE PREDICTION	41
3.1	COMPONENTS	42
3.1.1	SBR COMPONENT	43
3.1.2	APPOINTMENT CONTROLLER	43
3.1.3	PLAN-STEP CONTROLLER	44
3.2	COMPONENTS INTEGRATION	45
3.3	EXPERIMENTS	50
4	AUTOMATIC GENERATION OF PLAN LIBRARIES	53
4.1	PARAMETERS	53
4.2	PLAN LIBRARY GENERATOR	57
4.3	INPUT SET GENERATION	59
4.4	EXPERIMENTS	62
5	TEMPORAL MULTIPLE-PLAN RECOGNITION	69
5.1	ARCHITECTURE OVERVIEW	69
5.2	PLAN EXECUTION CONTROLLER	74
5.3	DATASETS	79

5.4	TEMPORAL PLAN LIBRARY GENERATOR	82
5.5	TEMPORAL INPUT SET GENERATION	85
6	EVALUATION	89
6.1	BASIC EVALUATION MEASURES	89
6.2	PLAN SEQUENCE AND SET OF OBSERVATIONS	94
6.3	USING A REAL-WORLD DATASET	96
6.4	SIMULATION RESULTS FOR SUBJECT ONE	99
6.5	SIMULATION RESULTS FOR SUBJECT TWO	101
6.6	SIMULATION RESULTS USING GENERATED PLAN LIBRARIES	106
7	RELATED WORK	111
8	CONCLUSION	115
	REFERENCES	117
	APPENDIX A – SUBJECT ONE – TABLES AND GRAPHS	129
	APPENDIX B – SUBJECT TWO – TABLES AND GRAPHS	139
	APPENDIX C – GENERATED PLAN LIBRARIES – TABLES AND GRAPHS	149

1. INTRODUCTION

One fundamental cognitive capability, critical to interpersonal interactions, is the ability to recognise plans and goals of other agents. This ability enables humans to reason about what other people are doing, why they are doing it, and what they will do next. Research that attempts to provide computers with comparable intention-recognition capabilities are variously referred to as activity recognition, plan recognition, intent recognition, and goal recognition. That way, the interest in *Artificial Intelligence* (AI) techniques to recognise humans' intentions and assist them to deal with real-world problems has increased in the last years [BBF15]. Research within the software agent community has focused on software assistants that can plan and act together with humans in realistic environments [Syc02, DMG⁺16], motivating the development of software assistants, in *Multi-Agent Systems* (MAS), which can operate on behalf of the user, giving advice to their human counterparts towards the completion of some plan of interest [GHK99, BFA00].

Nowadays, research in this area has been stimulated by the ubiquity of small, portable and powerful devices that offer significant computational capabilities. Thanks to advances in microprocessors, memory chips, smart sensors and network technologies, the ease of collecting observations from a wide range of sources has increased considerably, increasingly allowing everyday objects to have unexpected and innovative interactions [UI00, WADX15]. Many of today's electronic devices that people carry and interact with are faster, more efficient, cheaper and smaller due to these advances. These devices integrate several sensors that can generate a continuous stream of environmental data. Some examples of these technologies are [Yan09]: Wi-Fi networks, *Global Positioning System* (GPS), *Radio-frequency tags* (RFID), smartphones, etc. Since an agent can rely on smart devices technology for rich data collection, there is a growing requirement for automatic techniques that enable the correct analysis and treatment of all this data and deliver the right information to the appropriate agent at the right time, so that the agents can make better and faster decisions.

Techniques from AI combined with pervasive computing greatly increases the software assistants efficiency to pervasive assisted living making it more capable of providing a better quality of life in a non-intrusive way. Activity and plan recognition are important aspects in building pervasive environments, so recognising and understanding the activities and plans of people, through information provided by devices in the environment, is an important task in ubiquitous computing. However, this problem has been mostly tackled under the assumption that users carry out single activities one at a time and that activities executed consecutively all relate to a single plan, even though multitasking is generally an inherent characteristic of daily real-world routines as shown by [HHPZ⁺08].

Therefore, in this thesis, we aim to address these important features that are missing in most current research, as we discuss in Chapter 7. Furthermore, there is a lack of approaches with the ability to take temporal knowledge of each activity into account to predict anomalies in plan execution. Thus, in this thesis, we address the topic of activity and plan recognition for an observed agent carrying out multiple concurrent or interleaving activities in environments where each activity can have complex temporal dependencies. Our approach handles information about activity execution duration, i.e., how long is the interval of time in which the activity is normally performed, as well as the usual times of the day when the activities take place.

1.1 Motivation

One of the ultimate goals of Artificial Intelligence is to exploit the environmental and contextual signal in order to predict agent's intentions and information needs. Nowadays, techniques for human activity and plan recognition are employed in a wide range of scenarios, such as, advanced health-care delivery [LXAR13, ASSC02], improved assistance for elder people [KP14, PBC⁺03], traffic control [LWY14, AMC07], and smart homes [WHHB15, AN06]. The motivation for this work is the lack of systems to facilitate automatic monitoring and supporting agents performing concurrent activities, because most of the work in the literature assumes that users carry out single activities one at time. Furthermore, most of them cannot cope with all signals provided by the environment because they assume that an agent interacts with a single device at time. Besides, the majority of approaches can only treat binary signals from these devices. So, most previous investigations leave several open key challenges in these systems to assist an agent performing multiples plans in realistic environments, such as:

- Systems should be able to learn, understand, and automatically build a model of the agent's activities in a specific environment through observing what the agent usually does when performing a given plan;
- Systems should be able to monitor the agent's current activity by using learned knowledge, and alert the agent if it detects any deviation from the normal activity patterns that can lead a plan to fail;
- Most previous developed systems typically do not use information about execution duration of activities. Therefore, it can be interesting to develop mechanisms to learn and explicitly reason about such duration, and for example, estimate minimum and maximum duration for each activity. This information can be used to detect temporal inconsistency in activity execution, allowing the systems to alert the agent about a possible plan failure;

- Most previous work cannot cope with agents that pursue multiple concurrent plans, because they admit multiple plans only in sequence, where the observed agent can only move to another plan after it has finished a series of activities that are sufficient to recognise the current plan. In some domains, sequential plans are possible and common, but in many others, especially in real-world domains, agents can perform concurrent or interleaving plans, more importantly, with durative actions and the presence of multiple resources in a pervasive scenario, agents can effectively achieve multiple goals by executing multiple plans concurrently;
- In real-world domains, the similarity among plans execution and the incomplete knowledge about agent's intentions generate multiple candidate plans consistent with the observations. Most approaches are not able to disambiguate multiple hypotheses to find only the correct plan(s) being performed by the agent. Ambiguity in plan libraries based on real-world data is rather significant, as we show in one of our experiments.

Individuals often pursue several activities in a concurrent or interleaving manner in real-world situations, e.g., it is common for a person to answer the smartphone and watch TV at the same time, concurrently. So, if concurrent and interleaved plans are taken into consideration, we can potentially improve the recognition precision of many real-world plan recognition applications. Moreover, detecting anomalies in the duration of activities can provide an important evidence to an alert system. Since the system can capture the normal patterns in duration spent for each activity, it can also be used to detect an anomaly or a possible failure in a plan sequence. The study of several approaches addressing activity and plan recognition allowed a better understanding of the problems and challenges to develop a temporal multiple-plan recognition and find ways to deal with these problems. This study is important for applications requiring failure prediction, where the system developed must observe the agent, anticipate its next actions, and act pro-actively to give support in time, or even before, it is needed.

1.2 Objectives

An important and very difficult problem to solve in building pervasive and smart environments is how to learn and recognise user activities and goals, because in real-world circumstances, the users often pursue many activities in a concurrent and/or interleaving manner. Despite that, a majority of the approaches to activity and plan recognition focus on representing and learning sequential and temporal characteristics in activity sequences, assuming that the user carries out only one activity at time. So, we face the open challenges which need further study in modern activity and plan recognition (especially when considering realistic environments), such as:

- Develop an effective and accurate recognition method, able to handle interleaved plans (i.e., an agent may interrupt a plan to execute another, only returning to the remaining activities of the previous plan later) and/or concurrent plans (i.e., an agent carries out several plans that are pursued at the same time period);
- Consider temporal information on activities in order to demonstrate that this information improves plan recognition performance on real-world activities performed in smart environments, besides enabling detection of possible failures in a particular plan sequence;
- Develop a method able to deal with plan execution duration constraints (e.g., estimating the minimum and maximum duration for each activity) in the context of the Symbolic Behaviour Recognition (SBR) algorithms, allowing an efficient control of all recognition hypotheses consistent with the observations, taking into account concurrent plans and time/duration constraints;
- Create an efficient method able to recognise sequential, parallel and interleaved plans in realistic environments, dealing with the ambiguity in multiple hypotheses about the agent's plan(s) that are consistent with the observations and returning only plan(s) actually performed by the agent.

1.3 Thesis Organisation

The thesis is organised as follows: Chapter 2 provides a theoretical background about the main concepts addressed in this work. After that, in Chapter 3, we present a first version of the failure-prediction algorithm based on monitoring sequence of actions and action duration. The algorithms to automatically generate plan libraries and experiments testing the SBR time performance are shown in Chapter 4. The temporal multiple-plan recognition architecture as well as the real-world domain used to test this recogniser are explained in Chapter 5, along with the algorithms modified to create the temporal plan library, the temporal input set generator, and the algorithms developed to extend the SBR, able to deal with ambiguity, concurrent plans and temporal plan-recognition. The experiments performed in real and generated datasets to evaluate our approach and compare with SBR results are presented in Chapter 6. In Chapter 7 the related work is presented. Finally, Chapter 8 concludes and discusses future work.

2. BACKGROUND

This chapter presents the theoretical background that is the basis for this work. Introducing activity and plan recognition concepts, besides addressing issues related with the knowledge base representation, symbolic plan recognition, multi-agent plan recognition and ambient assisted living.

2.1 Activity Recognition

Activity recognition is the term used to describe the problem of identifying, or labelling, a series of single, possibly composite, activities performed by one or more subjects, according to a sequence of observations, or evidence [GG09]. Activity and plan recognition are distinct processes, while activity recognition focuses on a single activity, dealing directly with noisy low-level data gathered by physical sensors, plan recognition concentrates on identifying high-level complex goals by exploiting relationships between a set of activities that are elements of a complete plan [GG09, SGB⁺14].

An activity recognition system monitors an agent behaviour and its context in order to analyse and infer the underlying activities [CK11]. Basically the process can be described in the following steps. First the activity models that describe the possible performed activities in the environment must be created. Then, using sensors that acquire the contextual data, such as: cameras, wearable sensors, and instrumented user interfaces. The recogniser must monitor and capture the low-level user behaviour in order to process it and generate a high-level abstraction of the context to be used in the next step, in other words, the recogniser must discover and extract interesting patterns in noisy sensory data that can be interpreted as meaningful activities. Finally, an activity recognition algorithm is used to identify and encode the agent actions.

Observations may vary in nature depending on the activities and on the phenomenon under analysis. Different activity recognition approaches describe how observation attributes, or sets of features, contribute to identifying a particular activity in a sequence, the sequence is normally performed over time. Usually symbolic plan recognition systems use the actions performed by the observed agent directly as input assuming complete observability of the agent actions. On the other hand, in partially observable environments in which, for example, an agent is being obstructive, trying to hide his/her actions from the observer, some different kind of reasoning to infer unobserved actions might be needed. Geib and Goldman [GG01b] provide two kinds of reasoning for this type of environment. The first one takes into account the fact that actions might have predecessors that must be previously executed providing evidences of unobserved actions when an action is observed and the predecessors are not. The other reasoning

considers actions effect, since the effect of an action is more difficult to hide than the action itself, a state change can be used to infer that an action has occurred previously.

The observation process can be seen as a two-step process. First, there must exist a component responsible for gathering environment observations, then, these observations must be modelled to be used as input to the recogniser. Most plan recognition systems use experimental data provided by simulations and do not test their approaches in realistic scenarios with real users. To identify the context where the user is inserted, an application needs to monitor it to provide useful observations to the recogniser. The activities done by the user provide important contextual information. Real-world scenarios may use activity recognition to recognise actions and correctly encode it (i.e., in a first order sentence) to be properly used by the plan recognition system. Activity recognition also has been explored for physical phenomena, such as abnormal environmental events [HZY⁺09, YHY09] and for artificial entities, such as robots [LFH⁺03, VVL07]. A survey of techniques for human activity recognition and other applications can be found in [ASSC02, AMC07].

2.2 Plan Recognition

Plan recognition is the term used to describe the problem of inferring one or more subjects' goals based on a set of observed activities by constructing a plan (or multiple plans) that contains them [KA86]. Plan recognition research was initially defined in [SSG78] using a rule-based approach, like other early work such as [SA77]. Kautz and Allen [KA86] developed one of the first logical formalisation of plan recognition, providing the conceptual framework for much of the work in plan recognition up to the present. They defined the problem of plan recognition as finding a minimal set of top level actions adequate to justify the set of observed actions. Charniak and Goldman [CG93] argued that plan recognition is largely a problem of inference under conditions of uncertainty, and in addition to retrieving explanatory plans, a plan recognition system must also be able to select a hypothesis based on the likelihood of the explanation given the evidence.

Plan recognition systems can be classified according to the role that an observed agent plays in the plan recognition process [Wae96] into intended, keyhole, and obstructed plan recognition. In intended recognition, the observed agent is aware of the recognition process and actively participates giving signals to be sensed by the observer. An example of intended plan recognition can be found in a cooperative problem-solving environment. In keyhole recognition the user is unaware of the recognition process, which may provide only partial observability to the observer bringing the need to identify the context on its own. In a UNIX console, for example, users enter a sequence of commands to achieve a higher-level goal; an agent should infer the user's intention by observing the entered commands. In this case, the agent is following a keyhole plan

recognition process. Finally, the other position that an agent might assume in a plan recognition process is obstructive. In this case the user is aware of and obstructs the plan recognition process. This is a common user role in adversarial games, military and anti-terrorism scenarios. Obstructed plan recognition is applicable, for example, in a chess game, in which participants do not want their opponent to recognise their intentions, and do their best to hide them.

Since plan recognition is the process of inferring an agent’s plan, based on observations of its interaction with its environment, a plan recognition system must have a mechanism that is capable of inferring agent intentions by observing the agent’s actions in the environment. Thus, this mechanism, from a given set of observations, retrieves one or more hypotheses about the agent’s current plan of action. The knowledge used by this mechanism to infer plans is domain dependent, and therefore is commonly specified beforehand for each specific domain. This domain dependent information is usually encoded as two kinds of inputs for the recogniser: a sequence of actions from the observed agent, and a set of plans and goals. In other words, the inputs to a plan recogniser are generally a set of goals the recogniser expects the agent to carry out in the domain, a set of plans describing the way in which the agent can reach each goal, and a sequence of actions observed by the recogniser. The plan recognition process itself consists in inferring the agent’s goal, and determining how the observed actions contributes to reach it. The set of plans form a plan library and can include preconditions, effects, and sub-goals.

Symbolic plan recognition is a type of plan recognition mechanism that narrows the set of candidate intentions by eliminating the plans that are incompatible with current agent actions. Generally, symbolic approaches assume that the observer has complete knowledge of the agent’s possible plans and goals. Symbolic approaches handle the problem of plan recognition by determining which set of goals is consistent with the observed actions. Algorithms to recognise the intentions and plans executed by autonomous agents have long been studied in the Artificial Intelligence field under the general term of plan recognition. Such work has yielded a number of approaches to plan recognition [BST09, RG09] and models that use them in specific applications [GG01a, SS11, OMSN13]. Kautz and Allen [KA86] focus on symbolic methods providing a formal theory of plan recognition. Usually, these approaches specify a plan library as an action hierarchy in which plans are represented as a plan graph with top-level actions as root nodes, and plan recognition is then reduced to a graph covering problem, so the plan recognition process attempts to find a minimal set of top plans that explain the observations.

In a wide variety of applications, plan-recognition algorithms play a crucial role, such as human-robot interaction [CKS10], video surveillance [HN03, VA13], smart homes [AN06, KRSA13], intelligent user interfaces [UI00], and personal agent assistants [Gei02, SRS⁺16]. Plan recognition has been extensively used in many other areas

related or not with computer sciences including collaborative planning [HD93, KRS15], adversarial planning [AFH89, OnB15], story understanding [CG93, AZN98], discourse analysis [GS90], interface and collaborative agents [RS97, Bro98, LRS99], intelligent tutoring [GK95, KBdB⁺13], help systems [HBH⁺98], and games [AZN98, OnSU⁺13]. For a good overview of plan recognition in general, see Carberry [Car01], and for the most recent research in the field of plan, intention, and activity recognition, see Sukthankar et al. [SGB⁺14].

2.3 Knowledge Base

Most plan recognition systems require a knowledge base that encodes into recipes the ways in which agent goals can be achieved. A plan library is a knowledge base that codifies in some way the agent's beliefs concerning how the agent can reach each particular goal in the domain. Plan recognition systems have a plan library as an input, so several representations of agent plans have been used to approach this problem, and various methods applied to infer the agent's intention. These methods can be grouped in two main categories: symbolic and probabilistic approaches. Symbolic approaches aim at narrowing the set of candidate intentions by eliminating those plans that cannot be explained by the actions that the agent performs. The most used representation for symbolic approaches are plan hierarchies and consistency graphs. Probabilistic approaches explicitly represent the uncertainty associated with agent plans and allow a probabilistic ranking of the agent intentions mainly making use of Bayesian networks [Pea88] and Markov models [BVW02, BPV04]. Most symbolic and probabilistic approaches are domain independent and can lead to accurate predictions provided the plan library is complete (for symbolic approaches) or provided the probabilities are correct (for probabilistic approaches). These approaches normally have the disadvantage of considering all the possible plans in the plan library given the observations. However, if observations so far cannot distinguish between a set of possible intentions, probabilistic approaches can find the most probable one, while symbolic approaches can not select between them and have to wait for a single consistent explanation. Symbolic approaches are very sensitive to noisy actions, as the plan recogniser could wrongly exclude a plan (from the hypotheses explaining the observed behaviour) if an unexpected action occurs in the middle of the execution of a plan.

Many plan recognition systems structure their plan libraries as an Hierarchical Task Network (HTN) [EHN94a, EHN94b] to define the set of plans they are expected to recognise, in which goals are the root nodes and the observed actions are directly mapped to the leaf nodes. An attachment point in an HTN tree is a point in which an observation can be assigned to an action not observed yet, while shared leaders are action prefixes in the plan library that are common to different plans with different

goals (root nodes). Typically, a plan library has a single dummy root node where its children are top-level plans and all other nodes are simply plan steps. In the library, sequential edges specify the expected temporal order of a plan execution sequence and vertical edges decompose plan steps into sub-steps. The library has no hierarchical cycles. However, plans may have a sequential self-cycle, allowing a plan step to be executed during multiple subsequent time stamps. Each agent action generates a set of conditions on observable features that are associated with a plan. When these conditions are met, the observations match a particular plan step.

A complete algorithm for plan recognition must consider all coherent explanations for a given set of observations. A way to assess the domain complexity is to count the number of explanations for each observation, so Geib's [Gei04] main goal is to find out properties from a plan library that result in a large number of possible explanations. Kautz and Allen [KA86] describe the system knowledge about agent actions in terms of a hierarchy of specialisation and decomposition events, where decomposition captures the sub goals that are specific to an action and specialisation represents the different ways of performing a more general action. They specified the plan library as an action hierarchy in which plans are represented in a plan graph with top-level actions as root nodes, and plan recognition process is then a problem of graph covering. McCarthy's circumscription [McC86] was used to transform the hierarchy by circumscribing the ways of specialising an act, and then circumscribing the ways of using an act, closing the action hierarchy. From the action hierarchy, a set of axioms from which all desired conclusions could be derived deductively is specified. Brown [Bro98] uses Bayesian networks to build an agent profile that allows the detection of its intentions considering the utility of offering assistance. This approach represents the causality between preconditions, goals, and actions in terms of an AND/OR graph composing a goal hierarchy to represent user intentions. Goldman, Geib, and Miller [GGM99] assume that a plan library is made up of tasks structured in hierarchical way, in which task nodes could represent goals, methods, and primitive actions. Similarly to Brown [Bro98], the plan library could be viewed as a partially ordered AND/OR tree, in which the AND nodes are methods, connecting all action steps or sub-tasks needed to achieve the parent task, and the OR nodes are other isolated sub-tasks.

2.4 Symbolic Plan Recognition

The Symbolic Behaviour Recognition (SBR) approach [AZK05], briefly described below, is a method for complete, symbolic plan recognition that uses a plan library, which encodes agent knowledge in the form of plans. SBR extracts coherent hypotheses from a multi-featured observation sequence using a Feature Decision Tree (FDT) to efficiently match these observations to plan steps in a plan library.

2.4.1 Plan Library

A plan library is represented by a single-root directed acyclic connected graph, which includes all possible plans that an observed agent may execute. The term plan is used here in a broader sense, representing behaviours, reaction plans, and recipes. Typically, a plan library has a single root node in which its children are top-level plans and all other nodes are simply plan steps. Furthermore, in a plan library, sequential edges specify the expected temporal order of a plan execution sequence and decomposition edges decompose plan steps into alternative sub-steps. The plan library has no hierarchical cycles. However, plans may have a (sequential) self-cycle, allowing a plan step to be executed during multiple subsequent time stamps. Each agent action generates a set of conditions on observable features that are associated with that action. When these conditions are included, the observations match particular plan steps.

Figure 2.1 shows a plan library example presented in [AZK05], based on the behaviour hierarchies of Robocup soccer teams [KT00]. This plan library shows sequential links represented by dashed arrows and decomposition links represented by solid arrows. For instance, there is a sequential link between *attack* and *score*, and a decomposition link between *attack* and *pass*. The top-level plans are *defend*, *attack*, and *score*. Figure 2.1 does not show the set of conditions on observable features associated with plan steps, and circled numbers denote time stamps (e.g., *turn* has been considered a hypothesis at time stamp 2). The decomposition edges are shown only to the first child plan. Thus, in the figure, the path *root* → *attack* → *position* → *turn* → *without-ball* can be a hypothesis for the current plan being executed by an observed player.

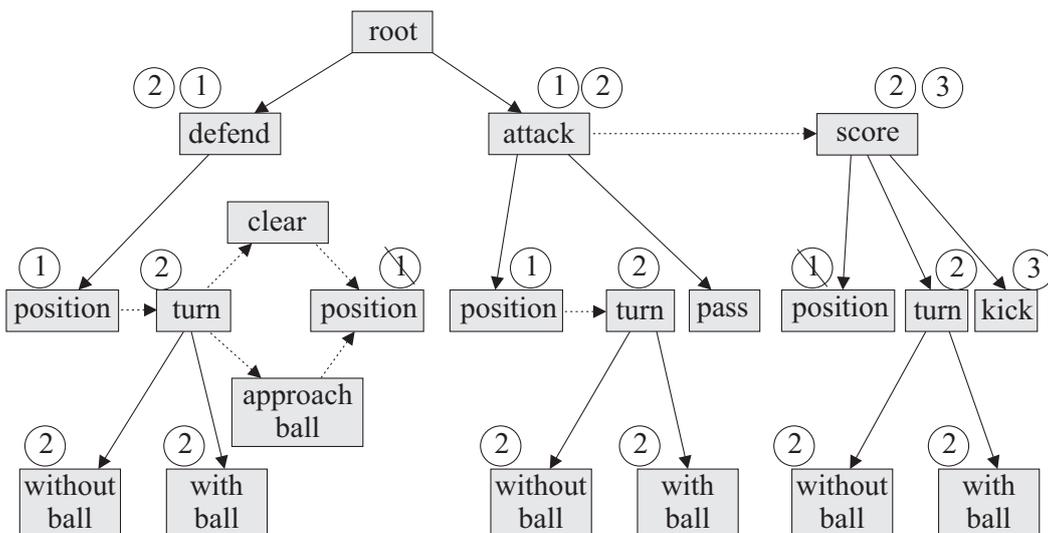


Figure 2.1: Plan library example of robocup soccer teams [AZK05].

2.4.2 Feature Decision Tree

The first stage of the SBR approach is the matching phase, in which the observations made by the recogniser are matched against plans in the plan library. The SBR algorithm considers complex observations presuming that each plan step has a set of conditions on observable features associated with it. When these conditions hold in regards to observed features of action execution (and in the correct order in case of sequential edges), the current observation is said to match that plan. Matching observations to plans can be computationally expensive if all plans are checked and for each plan all observed features are also checked (e.g., [KT00]). To speed-up this process of matching observations to plans, SBR augments the plan library with a feature decision tree data structure, which efficiently maps observations to matching nodes in the plan library. A FDT is a decision tree, where each node represents an observable feature and each branch represents one possible value of this feature. Determining all matching plans from a set of observed features is achieved by traversing the FDT top-down until a leaf node is reached. Each leaf node is a pointer to a plan step in the plan library.

The connection between FDT and plan library is shown in Figure 2.2, which shows part of a FDT using features associated with plan steps in Figure 2.1. A plan step executed by an agent can be identified according to observed features, and these features values are used by the FDT to separate the plan steps. To determine matching plan steps, the matching algorithm first checks FDT root nodes, which in Figure 2.2 are represented by the *have_ball* feature. After, the algorithm (based on feature value) continues the appropriate branch to test in sequence other features until a leaf node is reached. Thus, each leaf node will have pointers to all instance of the plan steps associated with it in the plan library [Avr09].

2.5 Multi-Agent Plan Recognition

Activity and plan recognition formalism generally assume that there is only one agent of interest. However, in many realistic environments, multiple agents are cooperating to perform a group task or they are simultaneously performing actions in the same environment. Action interdependence can arise due to the presence of multiple agents, so, this fact needs to be accounted for in order to perform accurate recognition. A single agent plan recogniser monitors an agent behaviour in order to match the observed actions with its knowledge about the agent intentions and plans. Although many of the issues related to the recognition process remains the same, Multi-Agent Plan Recognition (MAPR) must identify which agents are contributing to each particular plan and how their activities are connected.

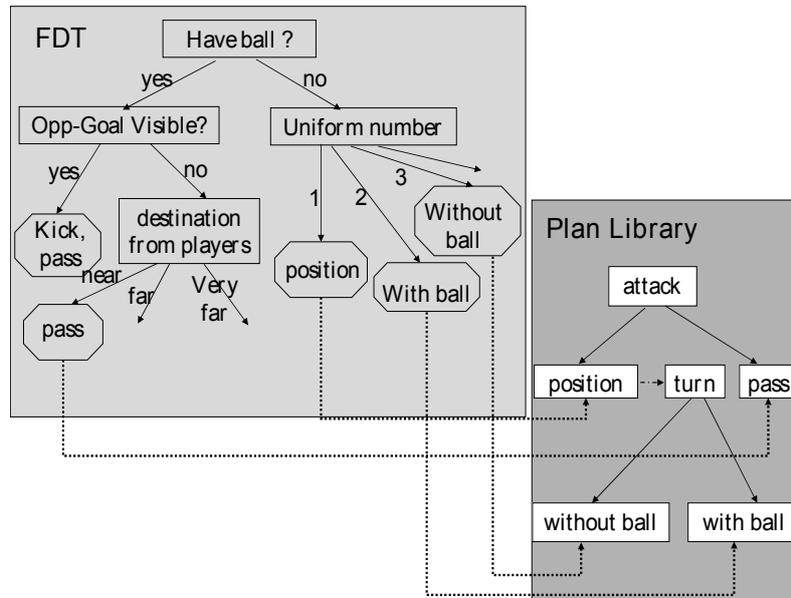


Figure 2.2: Example of a FDT associated with a plan library [Avr09].

Multi-agent plan recognition attempts to identify the dynamic team structures and behaviours from team observations, using a plan library that encodes known-team activities [ZL11]. One of the ways of differentiating MAPR approaches is by considering the way that the plan library is constructed or related to the recogniser. The simplest case for plan recognition in multi-agent scenario is the one in which agents from a unique team execute plan steps from a single plan library [IB99]. In these cases the plan recognition can be seen as concatenating individual agent hypothesis and matching them on the team plan library. Dynamic agent teams, in which the team members constantly change, represent a more complex scenario. Sukthankar and Sycara [SS08] propose a multi-agent plan representation that encodes dynamic team membership in order to extract temporal and resource (e.g., number of agents needed for a plan step) dependencies for creating a reduced plan library for then efficiently perform plan recognition on multi-agent traces (tuples of observed actions from an agent's team).

Kaminka and Tambe [KT00] focus on monitoring and detection of social relationships failures between teams of agents. They propose a framework to constantly monitor and compare the expected and actual behaviours of the observed agents in order to detect possible failures, diagnosing the reasons. The agent expected behaviour is encoded into a knowledge base as a plan hierarchy in which each plan step has its own pre-conditions and terminations conditions. Using a plan recognition algorithm to reason about individual agent behaviours and generating individual hypothesis to be used by other algorithms to compare different agents hypothesis, identifying failures and searching for inconsistencies in order to diagnoses the failure. Unlike single plan recognition approaches in which the plan library size might determine the hardness of the recognition problem [Gei04], Multi-Agent Plan Recognition is hard irrespective of how compactly the library is represented [BKL10].

2.6 Ambient Assisted Living

Ambient Assisted Living (AAL) focuses on providing assistance to people primarily in their natural environment [CCS⁺17]. Monekosso et al. [MFRR15] define AAL as “the use of information and communication technologies (ICT) in a person’s daily living and working environment to enable them to stay active longer, remain socially connected and live independently into old age”. Research in the AAL field covers a wide range of topics, but one of the largest is human activity and plan recognition, with the objective of detecting and recognising activity, plans, and goals within an environment.

Nowadays, ambient intelligence is increasingly used in order to support people in their everyday life. The increasing availability of sensors and intelligent objects enables new functionalities and services. In the AAL domain, such technologies can be used for monitoring the user behaviour, and reasoning about it to detect possible anomalous situations [MPPS17]. The AAL systems go beyond observing to interact with users, they use a variety of sensors for activity and plan recognition, sensing indirectly through environmental sensors or directly via wearable sensors and are able to make inferences about the cognitive or physical status of the user by analysing such streaming data. The analysis of the data can support automatic monitoring of numerous physiological and environmental data useful for identifying the current state and activity of the user.

Plan recognition can use very simple, unobtrusive sensors to reduce privacy concerns and increase acceptance by users. Among the variety of sensors that can be applied, there are personal mobile devices, such as smartphones equipped with motion and location sensors. Besides, wearable sensors are able to collect data from physiological signals or data reflecting the body movement. Most of these sensors measure environmental parameters, deducing human plan from observation. The rise of mobile computers and pervasive computing (also called ubiquitous computing), with devices providing different services in different contexts, will enable us to find novel solutions to help people in their everyday life. Pervasive computing then evolved from providing functionality transparently to adapting functionality to the user’s context, whereas the most modern applications are increasingly using information from the user’s current context to pro-actively adapt to the user’s needs [RBBG10].

The pervasive presence around us of a variety of things or objects that are able to interact with each other and cooperate with their neighbours to reach common goals is the basic idea of Internet of Things (IoT) [WADX15]. The main strength of the IoT idea is the high impact it will have on several aspects of everyday life and behaviour of potential users [AIM10]. The impact of the IoT introduction, from the point of view of a private user, will be evident in both working and domestic fields [Sat01]. AAL is one of the typical applications of IoT, which aims to support independent living, motivated by the worldwide trend towards an ageing population [FXL⁺17].

Final Remarks

Along the studies to support this thesis, some works were performed in order to auxiliary on development and elaboration of ideas and mechanisms to face the open challenges in modern plan recognition and AAL. An initial work to create a failure predictor, based on monitoring sequences of actions and action duration, was developed using SBR to find out the plan that is being performed by an observed agent, addition to using a plan library to represent the knowledge base about the activities sequence that can be performed in the environment, this work is presented in next Chapter 3.

3. FAILURE PREDICTION

Recently, the number of real-world applications that deal with the need to recognise goals and plans from agent observations is on the rise. These applications can be found in fields such as human assisted living [Gei02, CCAY13], interface agent systems [AA07], human-computer interaction [Hon01, RA15], military scenarios [MG04], story understanding [CG90], traffic monitoring [PW95, HLR⁺13], and others. However, techniques that include the task of anticipating failures during agent plan execution have received little attention. Multi-agent environments are dynamic since they are in a constant state of change resulting from agents' actions. When these changes occur, a plan that was expected to work before may fail, so anticipating, from agent observations, when a plan is going to fail can be an important mechanism during the plan recognition process. Commonly, plan recognition approaches cannot make such inferences, which means that when an agent has no intention to complete or finish a plan, these approaches will continuously attempt to recognise what the agent is doing.

In daily activities most people interrupt the plan that they are performing for some reason, such as, getting their attention drawn to something else, getting distracted by other events, or being interrupted by an emergency that needs immediate attention. In plan recognition context, we consider that a plan is going to fail when the sequence of actions is taking too long or does not match the plan which the observed agent should be performing at the moment. Thus, our initial approach uses a calendar for managing some of the agent's goals over the near future, and when that information is available it facilitates our failure checking procedure, as well as plan recognition disambiguation. A plan failure can occur when an agent interrupts its current plan execution due to concurrent plans that need attention, or when an agent has to deal with conflicting plans. In this case, a mechanism to anticipate failures during agent plan execution can be useful in several situations, for example, helping an agent stay focused on a particular plan, or detecting when an agent is deviating from its regular activities.

Research on planning has focused on the modelling of actions with duration and stochastic outcomes, both theoretically as variants of Markov Decision Processes (MDP) [MW08], and domain description languages that express temporal planning (e.g., PDDL 2.1 [FL03], an extension of PDDL). In the literature, a similar approach to failure prediction is plan abandonment detection, Geib and Goldman [GG03] propose a formal model to recognise goal/plan abandonment in the plan recognition context, based on the Probabilistic Hostile Agent Task Tracker (PHATT) [GGM99]. This formal model estimates the probability that a set of observed actions in sequence contributes to the goal being monitored. Furthermore, Geib [Gei02] addresses some issues and requirements for dealing with plan abandonment, as well as intention recognition in elderly-care domain.

In this initial work, we develop an approach for predicting plan failure by monitoring agent's actions during its plan execution [FPH⁺16, FPH⁺17]. Essentially, our approach to plan failure prediction features a mechanism that is composed of three modules. First module is responsible for recognising the plan that the observed agent is executing. Second module checks if plans assigned to observed agent are being executed as scheduled in a predefined calendar. Lastly, the third module checks if actions are being executed as expected (i.e., not taking too long, and matching the current monitored plan). Thus, this approach can be used in complex software system, including health-care applications, to improve functionality, such as, activity recognition and task reallocation [PFS⁺15] among agents representing human users who collaborate to take care of a patient, by detecting if a person responsible for some activity of the patient is following his scheduled appointments; detecting problems that may prevent the person in charge to attend to his obligations and send warning to the system.

An agent can attempt to achieve multiple goals and each goal can be achieved by applying various different plans. Anticipating failures in agent plan execution, before they occur, is an important mechanism to enable the agent and the overall system to elaborate strategies to avoid or circumvent them, allowing the goal to be achieved successfully. Plan recognition is used to infer which plans are being executed from observations of sequences of activities being performed by an agent. The Symbolic Behaviour Recognition (SBR) algorithm (presented in Section 2.4) represents knowledge about the agents under observation in the form of a plan library. We use SBR to find out which plan the agent is performing and we develop a failure prediction system, based on information available in the plan library and in a simplified calendar that manages the goals the agent has to achieve. This failure predictor is able to monitor the sequence of agent actions and detect if an action is taking too long or does not match the plan that the agent was expected to be performing.

3.1 Components

The failure predictor is responsible for predicting plan failures during execution of an agent goal, more specifically, it tracks the execution of a goal and attempts to identify elements which can lead it to fail, for example, an action taking significantly more time than expected to conclude. The failure predictor is composed of three modules: SBR module, responsible for recognising the plan being executed by the observed agent; Appointment Controller which checks if the goals that are known to have been assigned to the agent are being executed as scheduled; and Plan-Step Controller that checks if the plan steps (that compose the plan) are being executed as expected. These modules are presented, respectively, in Subsections 3.1.1, 3.1.2, and 3.1.3.

3.1.1 SBR Component

The SBR component implements the Symbolic Plan Recogniser presented in Section 2.4, responsible for recognising the plan that an agent is currently executing, and generating hypotheses about the possible plans while the recognition is still not possible. This information is represented both as a list of candidate plans as a hypotheses graph. The SBR component receives observations as input, *i.e.*, sets of contextual information about the observed agent and its actions. Examples of observations include the agent's global positioning coordinates, whether or not the agent is moving, or whether the agent is approaching a particular place, and any other contextual information that can be generated by an activity recognition process. As output, this component provides both the list of candidate plans and the hypotheses graph.

3.1.2 Appointment Controller

A plan library contains all known plans (agent goals) for a given domain, together with the set of actions that compose them, however, it does not define the time that the agent is expected to execute each plan, neither does it contain the time interval in which the plans have to be executed. These are essential information for the system to ensure that plans are being executed in an appropriate manner and to be able to detect potential failures in plan execution. The Appointment Controller component implements a simplified calendar which manages the agent goals and plans. It defines which plans of the plan library an agent is known to be responsible for, and the time an agent is expected to execute some of them (to the extent that this is known in particular domains). This component also helps in disambiguation of candidate plans and in the early prediction of plan failures. It should be noted that only domain-related plans are kept in this individual calendar. Each entry (*i.e.*, agent goal) in the Appointment Controller is composed of the following information:

starting date – date in which the goal or plan is expected to start;

ending date – date in which the goal or plan is expected to end;

title – title of the goal or plan (*e.g.*, “*managing-medication*”);

description (optional) – brief textual description of the goal (*e.g.*, “*take medicine*”);

related plan ID – unique identifier of the relevant plan (*i.e.*, the plan to achieve this goal), which corresponds to a top plan in the plan library (*e.g.*, “p1”);

tolerance – margin of error to start and finish each goal. For example, some goals can start or finish 5 minutes before (or after) the time it was originally scheduled without danger to provoke a plan failure (e.g., a “meeting” can start or finish a few minutes more/less than was previously scheduled). This tolerance is necessary because in real-world goals usually do not start/finish exactly at the scheduled time.

3.1.3 Plan-Step Controller

The Plan-Step Controller monitors and analyses the plan execution (actions sequences) to detect anomalies that can lead to plan failure. The necessary information to the Plan-Step Controller operating is obtained through the SBR component, providing information about the current plan and actions being performed; the plan library; and the file containing information about the expected execution time for each action. The plan library contains the known plans and the necessary actions for a given plan to finish successfully, besides the actions sequence that must be performed for a plan to be completed. However, it does not define when a plan should finish, neither the time in which the actions have to be executed. This information is important to detect anomalous behaviour during plan execution, such as:

plan interruption – plan execution interrupted without all actions completed;

time exceeded – an action takes significantly more time than expected, this typically leads to plan failure (e.g., being in a traffic jam);

inconsistent sequence – a sequence of observed actions is inconsistent with the expected plan path in the plan library.

It is important to keep track of the actions being performed in order to be able to predict whether a plan is following the expected execution path. Thus, it is possible to identify a probable failure in plan execution and generate the required warnings according to failure type. The data file with the expected execution time for each action is composed by the following information:

plan-step ID – unique identifier of the related action that corresponds to a plan step in the plan library (e.g., “p1.11”);

label – label for identification of the action (plan step) from the plan library (e.g., “taking-medication”);

time – time that an action often take to be performed;

tolerance – value by which the ending time of the action is allowed to be delayed. For instance, an action can take 5 minutes in addition to its normal time to be performed. This tolerance is important for real-world situation where actions can often take more time to be performed than an exact specified time.

3.2 Components Integration

The failure predictor components are integrated as presented in Figure 3.1. When the SBR is not able to determine the current plan (no plan or multiple plans were recognised) the Appointment Controller component is consulted (using the output of the plan recogniser). First, the component checks if there are plans scheduled for the moment in which it was consulted and, later, if a scheduled plan is in the list of candidate plans. During this verification the following situations may occur:

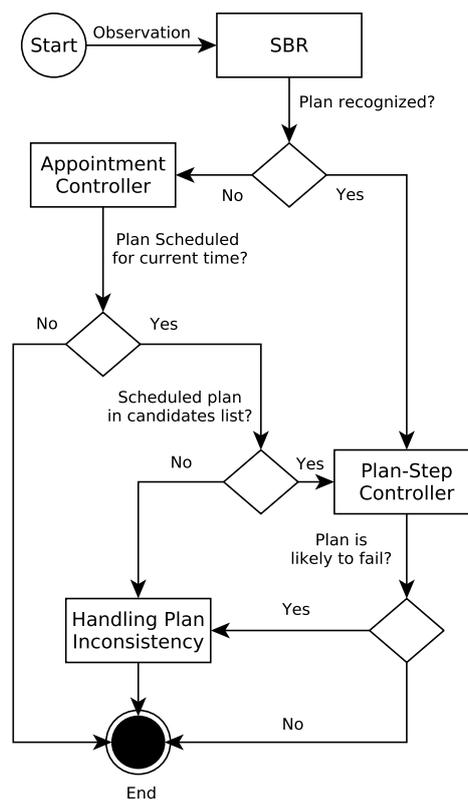


Figure 3.1: Integration of failure predictor components.

- There is no agent goal scheduled for the current time. In this situation, the Appointment Controller component has nothing to do, so the main cycle ends and the system awaits for a new observation.
- There is a plan scheduled for the current time. However, the candidate plan list is empty (no plan was recognised). In this case, Appointment Controller detects

a failure in the scheduled goal execution (i.e., a goal expected to be executed at the time the calendar was consulted). In this case, a warning must be sent to the system, which should be able to handle this plan failure.

- There is a plan scheduled for the current time and there are multiple plans in candidate plans list. In this case, the Appointment Controller verifies if the plan related to scheduled goal is present in plan candidate list. During this verification two situations can happen:
 - The plan linked to the scheduled goal is in the list of candidate plans. Thus, this plan is considered as currently being executed by the agent;
 - The plan related to the scheduled goal is not in candidate plans list. In this situation, a failure is detected in the scheduled goal execution as it is not being executed by the agent as it should. Then, a warning related to the scheduled goal is sent to the Handling Plan Inconsistency step of the main cycle, in which the system will handle the plan related to the failing goal.

Regarding goal scheduling in this initial implementation does not allow overlapping of goals (i.e., goals with coincident time intervals). That is, a new goal will not be added to calendar if it overlaps with an existing one. Given two hypothetical goals (*A* and *B*), they are considered as overlapping if one of the following is true:

- *A* starts and ends at the same time as *B*. For example, *A* (2:00 p.m. to 4:00 p.m.) and *B* (2:00 p.m. to 4:00 p.m.);
- *A* starts before *B* and *A* ends after *B* started. For example, *A* (1:00 p.m. to 3:00 p.m.) and *B* (2:00 p.m. to 4:00 p.m.), or *A* (1:00 p.m. to 5:00 p.m.) and *B* (2:00 p.m. to 4:00 p.m.);
- *A* starts after *B* has started, however, before *B* has ended. For example, *A* (3:00 p.m. to 5:00 p.m.) and *B* (2:00 p.m. to 4:00 p.m.).

However, there is an exception for the overlapping rule regarding the starting and ending times. Two goals are not considered as overlapping if the starting time of one of them is equal to the ending time of the other one. This exception is convenient, as sequential plans are usually scheduled with no time interval among them, e.g., *A* (1:00 p.m. to 2:00 p.m.) and *B* (2:00 p.m. to 4:00 p.m.) or *A* (4:00 p.m. to 5:00 p.m.) and *B* (2:00 p.m. to 4:00 p.m.).

The Plan-Step Controller is unable to disambiguate the list of candidate plans, thus both SBR and Appointment Controller must inform only one plan/goal (top-level node) and only one plan step (internal/external node) at each iteration. The *planController* (Algorithm 1) is responsible for handling such information, using the plan library and a

data file with plan step running times, in order to check the plan sequence (based on plan steps) and monitor the running time of each action. So, it can detect and report possible changes in plan execution avoiding possible failures.

Algorithm 1 *planController*(η_g, η_{ps}, l)

Input: Current Goal η_g , Current Plan Step η_{ps} , Time List l

- 1: check η_{ps} duration present in l
 - 2: *checkCurrentGoal*(η_g)
 - 3: *checkCurrentPlanStep*(η_{ps})
 - 4: Inform possible failure
-

The monitoring of current plan is performed by *checkCurrentGoal* (Algorithm 2). Initially, current goal is updated based on information received by SBR (Line 1) and a test is performed to check if it is a valid value (Line 2). After that, the algorithm checks if current goal is equal to previous goal (Line 5) which the agent was trying to achieve. If they are the same, it means agent is still carrying out actions to achieve it. Otherwise, the agent started to perform a new goal with a new plan. In this case, the algorithm has to verify if the previous goal was achieved successfully (Line 11). For that, the information in the plan library is used to check if the last plan step (of the previous goal) is a leaf node. If so, it means that the plan was fully executed and probably has finished successfully. Otherwise, the agent may have stopped performing the plan before its finish or the agent is executing more than one plan at the same time. Thus, the algorithm should send a warning about this possible failure (Line 15).

The *checkCurrentPlanStep* (Algorithm 3) is responsible for monitoring the execution of each action related to a goal, i.e., it analyses the sequence of execution and the runtime of each plan step. Initially, current plan step is updated based on information received by SBR. The consistency of this information is checked and the algorithm then checks if current plan step is equal to previous plan step which the agent was performing (Line 5). If they are equal, it means that the agent is still performing the same plan step. Thus, the algorithm has to check if current action is within the time specified in data file with plan step running times. The *checkExecutionTime* (Line 7) receives as input the current plan step and checks if its runtime is within the specified time, taking into account the specified tolerance for each action. If the agent is taking too long to perform some action, the algorithm detects this as unexpected behaviour and warns the system.

When failure predictor detects a new action being performed by an agent, it is necessary to check if this action is part of the sequence of actions needed to accomplish the current goal (Line 9). The *isValidSequence* (Algorithm 4) uses information in plan library to check if current plan step is part of a valid sequence of actions to achieve current goal, receiving as input the previous plan step and the current plan step. If current plan step contains a sequential parent node and this parent node is the previous

Algorithm 2 *checkCurrentGoal*(η_g)

Input: Current Goal η_g

```

1:  $\eta_g^* \leftarrow$  previous goal
2: if  $\eta_g = null$  then
3:   no goal can be checked
4: else
5:   if  $\eta_g = \eta_g^*$  then
6:      $\eta_g$  keeps running
7:   else
8:     if  $\eta_g^* = null$  then
9:        $\eta_g$  get started
10:    else
11:      if  $\eta_g^*$  has finished in an external node then
12:         $\eta_g^*$  has finished and it's completed
13:         $\eta_g$  get started
14:      else
15:         $\eta_g$  get started
16:         $\eta_g^*$  has stopped before its completed
17:      end if
18:    end if
19:  end if
20: end if

```

Algorithm 3 *checkCurrentPlanStep*(η_{ps})

Input: Current Plan Step η_{ps}

```

1:  $\eta_{ps}^* \leftarrow$  previous plan step
2: if  $\eta_{ps} = null$  then
3:   no plan step can be checked
4: else
5:   if  $\eta_{ps} = \eta_{ps}^*$  then
6:      $\eta_{ps}$  keeps running
7:     checkExecutionTime( $\eta_{ps}$ )
8:   else
9:     if isValidSequence( $\eta_{ps}^*, \eta_{ps}$ ) then
10:      current execution path is right
11:    else
12:      current execution path has changed
13:    end if
14:  end if
15: end if

```

plan step, it means that execution path is correct. Otherwise, current plan step does not match the execution path done so far to achieve current goal.

Detecting whether execution sequence is valid is more complicated when current plan step has a decomposition parent node, because previous plan step does not have to be a parent node of the current plan step, but only be part of current plan execution and follow the temporal order of the execution path. The *isPreviousNode* (Al-

Algorithm 4 $isValidSequence(\eta_p, \eta_c)$

Input: Parent Node η_p , Child Node η_c

```

1: if  $\eta_c$  has a sequential parent then
2:    $\eta_p^* \leftarrow$  sequential parent of  $\eta_c$ 
3:   if  $\eta_p^* \neq \eta_p$  then
4:     return false
5:   else
6:     return true
7:   end if
8: else if  $\eta_c$  has a decomposition parent then
9:    $\eta_p^* \leftarrow$  decomposition parent of  $\eta_c$ 
10:  if  $\eta_p^* = \eta_p$  then
11:    return true
12:  else
13:    if  $isPreviousNode(\eta_p, \eta_c)$  then
14:      return true
15:    else
16:      return false
17:    end if
18:  end if
19: else
20:   return false
21: end if

```

gorithm 5) algorithm receives as input the previous plan step and the current plan step, then it checks the entire running sequence from current plan step node to previous plan step node in order to analyse the temporal order to determine if current plan step is a valid sequence for current goal execution.

Algorithm 5 $isPreviousNode(\eta_p, \eta_c)$

Input: Parent Node η_p , Child Node η_c

```

1: if  $\eta_c$  has a sequential parent then
2:    $\eta_p^* \leftarrow$  sequential parent of  $\eta_c$ 
3:   if  $\eta_p^* = \eta_p$  then
4:     return true
5:   else
6:     return  $isPreviousNode(\eta_p, \eta_p^*)$ 
7:   end if
8: else if  $\eta_c$  has a decomposition parent then
9:    $\eta_p^* \leftarrow$  decomposition parent of  $\eta_c$ 
10:  if  $\eta_p^* = \eta_p$  then
11:    return true
12:  else
13:    return  $isPreviousNode(\eta_p, \eta_p^*)$ 
14:  end if
15: else
16:   return false
17: end if

```

3.3 Experiments

The objective of experiments is to show how our initial approach provides helpful reminders by monitoring and anticipating plan failure from agent observations. In this way, we model part of a scenario that represents agent behaviour based on Activities of Daily Living (ADL), which is a term used in health-care to refer to daily self-care activities of people. Figure 3.2 shows an example of plan library based on ADL, where the activities correspond to user single actions (e.g., *getting-up*, *watching-tv*, *reading-a-book*, *taking-medication*, *using-bathroom*). In this scenario, there is a person with disabilities who lives alone and needs constant monitoring to perform his daily activities. Using plan library formalism, we model a set of plans for representing possible behaviour of this person, where some of these plans are shown in Figure 3.2.

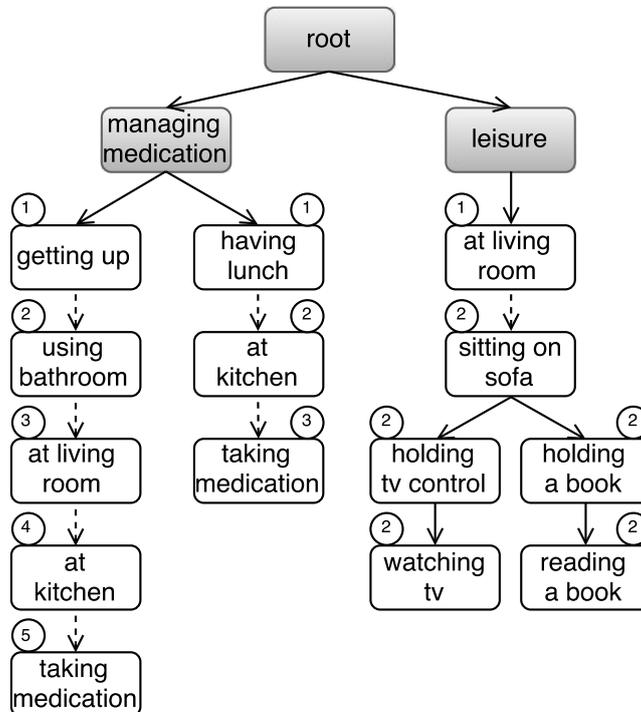


Figure 3.2: Example of a plan library based on ADL.

To exemplify how our initial approach works, we schedule the top-level plan *managing-medication* to be performed between 7:00a.m. and 7:30a.m., in which this schedule information is in the calendar (Appointment Controller). Considering the current part of day as early morning, the sequence of activities to be performed in order to accomplish the plan *managing-medication* is: *getting-up* → *using-bathroom* → *at-living-room* → *at-kitchen* → *taking-medication*. According to this sequence, the user must move through the living room (i.e., plan step *at-living-room*) to complete the plan. However, this plan step is also part of the top-level plan *leisure*, in this case, SBR component returns both top-level plans *managing-medication* and *leisure* when current plan step is

at-living-room. To deal with this ambiguity, our approach uses Appointment Controller component to check if there is a top-level plan scheduled for the current time, if so, it discards those plans that are not scheduled for this time.

```

1 ...
2 [Info]:Current Top-Level Plan: managing-medications
3 [Info]:Current Plan Step (PS): at-living-room
4 [Info]:Checking time [at-living-room]
5 [Info]:PS [at-living-room] started at 7:15am
6 [Info]:PS [at-living-room] is running for 5 min
7 [Info]:PS [at-living-room] avg time 3 min | tol 1 min
8 [Warn]:PS [at-living-room] is taking too long
9 ...

```

Listing 3.1: Example of failure predictor output.

The example of program output, presented in Listing 3.1, represents part of execution of the failure predictor initial approach, in a scenario where the user must take a medication in a strict time and during plan execution the user's attention drawn to something else (e.g., the user stays at living room watching TV) and forgets to take his medication. In this case, current top-level plan is *managing-medication* (Line 2) and current plan step is *at-living-room* (Line 3). The failure predictor monitors the plan execution (Lines 4-6) and based on information in the Plan-Step Controller, i.e., average time of execution for each plan step and a time tolerance (Line 7), it is able to detect anomalies in plan execution and informs the system to try to address and correct these possible failures. In this example, the algorithm detects that a plan step is taking too long to be performed, then an alert message is generated (Line 8).

Final Remarks

This chapter presented the first steps to developing a failure predictor based on plan recognition techniques and a calendar that includes *some* of the plans that the agent is known to be required to execute over time. The main contribution is a system that anticipates plan failures by monitoring a sequence of agent actions during its plan execution. An automatic generator of plan libraries for plan recognition performance evaluation also was developed with the aim of testing the performance of several plan recognition algorithms with different structures of plan library. Chapter 4 presents this work which enabled a greater understanding of how to create and represent complex structures to represent possible agent behaviour as plan library, as well as its advantages and limitations.

4. AUTOMATIC GENERATION OF PLAN LIBRARIES

Most approaches to plan recognition are based on manually constructed rules, where the knowledge base is represented as a plan library for recognising plans. Although such approaches use conceptually similar plan library representations, they seldom, if ever, use the exact same domain in order to directly compare their performance. For any non-trivial domain, such plan libraries have complex structures representing possible agent behaviour, so plan recognition approaches often fail to be tested at their limits and only rarely they are compared with each other experimentally, leading to the need for a principled approach to evaluating them. In order to address this shortcoming, we develop a mechanism to automatically generate arbitrarily complex plan libraries. Such plan library generation can be directed through a number of parameters to allow for systematic experimentation. We validate our mechanism by generating and carrying out experiments to evaluate the performance of the SBR algorithm.

4.1 Parameters

The algorithms developed in this work were created in order to enable systematic analysis and performance comparison between several plan recognition algorithms given the variety of possible plan libraries. Thus, the *Plan Library Generator* (Section 4.2) generates plan libraries based on various given parameters (summarised in Table 4.1), and the *Input Set Generator* (Section 4.3) generates sequences of legal observations, given a plan library [FHM⁺15].

Table 4.1: Parameters available to generate the plan library

Flag	Symbol	Default	Constraints	Description
-g	$\$g$	$\$g = 3$	$\$g \geq 1$	number of top-level plans
-d	$\$d$	$\$d = 4$	$\$d \geq 1$	depth of the plan trees
-b	$\$b$	$\$b = 1$	$1 \leq \$b \leq \B	minimum number of branches
-B	$\$B$	$\$B = 3$	$\$b \leq \B	maximum number of branches
-f	$\$f$	$\$f = 1$	$\$f \geq 1$	number of features per node
-c	$\$c$	$\$c = 2$	$\$c \geq 1$	number of values
-F	$\$F$	$\$F = 10$	$\$F \geq (\$d \times \$f)$	available values to the features
-s	Φ	$\Phi = 0.8$	$0 \leq \Phi \leq 1$	sequential edges probability
-D	Ψ	$\Psi = 0$	$0 \leq \Psi \leq 1$	duplication probability
-p	path	<i>plan_library.xml</i>	$\langle String \rangle$	output file path

- **Number of top-level plans** ($\$g$): represents the branching factor of the root node, in other words, the number of children for the root node. This value refers to

number of different independent top-level plans in a plan library, presented as gray circles in Figure 4.1.

- **Depth** ($\$d$): corresponds to the depth of plan trees. This depth value for the plan library (from the root node) determines the number of plan steps that a plan instance contains. In Figure 4.1 the number of nodes in a complete path from a top-level node to a leaf node is equal to $\$d = 3$.
- **Minimum number of branches** ($\$b$): defines the minimum number of branches that all nodes (other than root and leaf nodes) must have. This value must be in the interval $[1; \$B]$. In Figure 4.1 all nodes have at least one branch ($\$b = 1$).
- **Maximum number of branches** ($\$B$): defines the maximum number of branches that all nodes (other than root and leaf nodes) may have, and this value must be greater than or equal to the minimum number of branches ($\$b$). The number of branches is randomly chosen from the interval $[\$b, \$B]$ whenever a new node is created. In Figure 4.1 node ps2.1 has the maximum number of branches ($\$B = 3$).

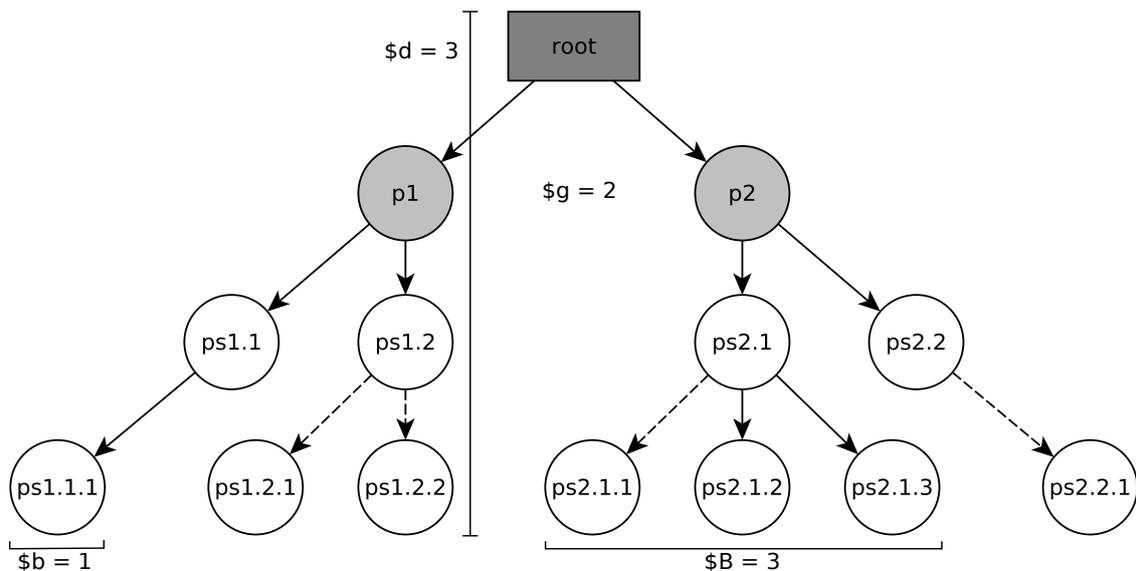


Figure 4.1: Example of a plan library with $\$g=2$, $\$d=3$, $\$b=1$ and $\$B=3$.

- **Number of features** ($\$F$): number of observable features, in the domain, available to be associated with a given plan step. Features are properties associated with the action represented by a given plan step, which need to be observed by the plan recognition algorithms for them to be able to recognise the execution of this particular plan step. In Figure 4.2, three observable features (f_1 , f_2 and f_3) may be associated with nodes representing top-level plans or plans steps.
- **Number of features per node** ($\$f$): number of features associated with each top plan and plan step. As restriction, the number of features ($\$F$) must be greater than (or equal to) depth ($\$d$) multiplied by number of features per node ($\$f$).

$$\$c \geq \$d \times \$f$$

In other words, there are at least $\$f$ distinct features for each plan step belonging to a single plan instance. In Figure 4.2, each node must have at least one distinct feature ($\$f = 1$) with an associated value to make possible the plan recognition process. In cases where a feature is not considered for the plan step recognition process we put a hash sign “#” to denote that this feature has no influence in the recognition of this specific plan step.

- **Feature status** ($\$c$): represents the number of values that may be associated with the features. Features with a specific status allow the identification of determined plan steps (actions) being executed by the observed agent. This value must be greater than or equal to one ($\$c \geq 1$). In Figure 4.2, for instance, considering feature status equal to two ($\$c = 2$) means that all features are multi-valued and get values equal to zero, one or two. It is worth noting that, in this case, only integer values in the interval $[0, \$c]$ can be assigned to the features.

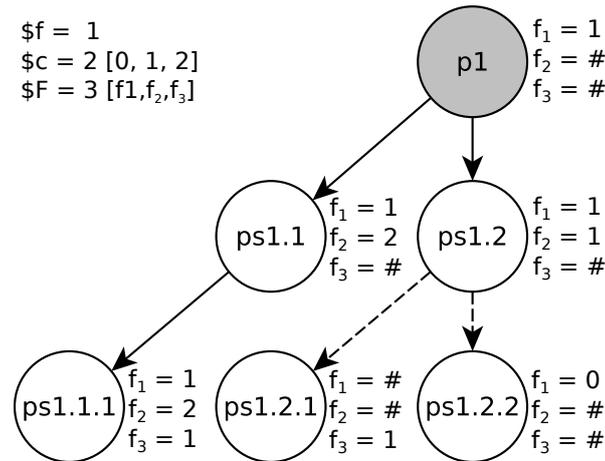


Figure 4.2: Part of a plan library with features valued $\$f=1$, $\$c=2$ and $\$F=3$.

- **Sequential edges** (Φ): value in the interval $[0, 1]$, which determines the probability of a branch to be created as sequential (rather than decomposition). Thus, for example, $\Phi = 0$ means that all branches will be of the decomposition type (Figure 4.3 (a)), whereas $\Phi = 1$ means that all branches will be sequential (Figure 4.3 (c)). Regarding feature distribution, it is important to emphasise that the same feature can not be assigned with different values to a node and its respective decomposition children because the children node represent a specialisation of the parent node. This way, when a decomposition node is created, it automatically inherits from its parent node the features as well as the values attributed to them.

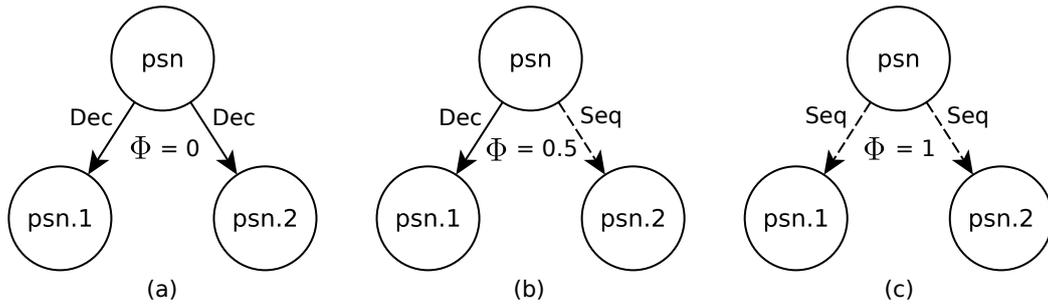


Figure 4.3: Examples of branch creation, solid lines represent decomposition links and dashed lines represent sequential links.

- Duplication (Ψ):** represents the percentage of top-level plans that are duplicated in order to generate ambiguous paths. Each plan in the plan library has a unique identification; however, plans that present the same set of associated features and the same value of these features are similar in the sense that they will match the same set of observations, at least up to a point hence leading the recogniser to have multiple unresolved hypotheses. The duplicated plan is not exactly equal to the top plan from which it was generated, because the last leaf plan step in a duplicated plan is created different to establish some distinction between them (Figure 4.4). For example, $\Psi = 0.2$ means that 20% of top level plans are a duplicate of another, thus approximately 40% of top level plans are not unique, having some difference in the leaf nodes.

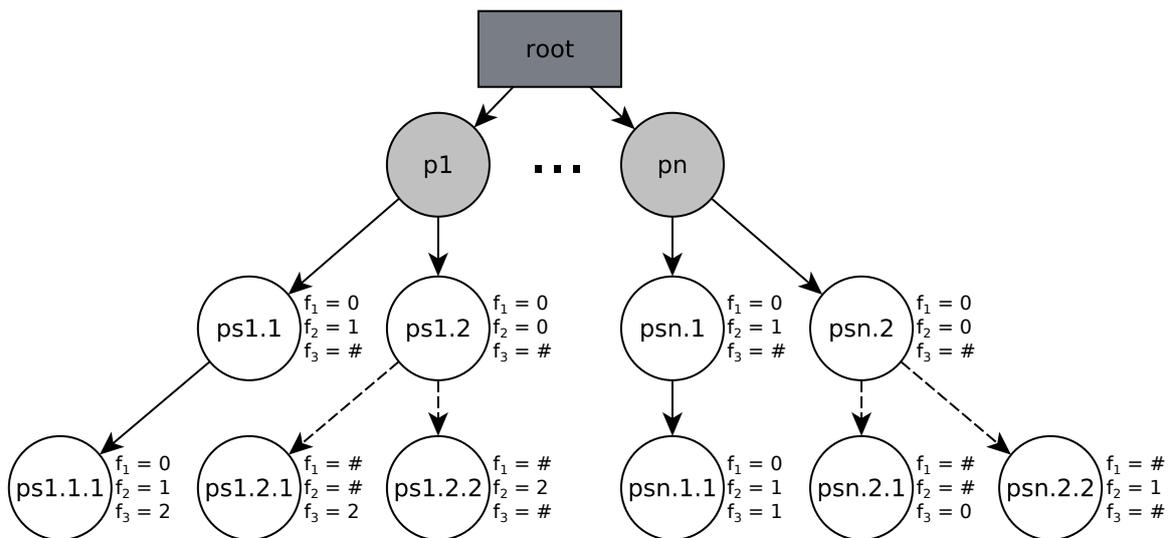


Figure 4.4: Example of duplication in a plan library where top-level plan p_n is a duplication of p_1 differing only in the leaf nodes.

4.2 Plan Library Generator

The plan library generation is conducted as shown in Algorithm 6. This algorithm starts creating the root node (Line 1), which is responsible for connecting all top-level plans/goals, being created as decomposition type with no features assigned to it. Next, the algorithm determines how many plans will be duplicated (Line 2) and the number of distinct top-level plans to be created (Line 3). For example, whereas that $g = 10$ and $\Psi = 0.1$ then we will have 9 distinct plans created and the last one will be basically a copy of a previously created top plan. This is important for evaluating plan recognition algorithms that typically keep track of sets of potential plans being executed until some disambiguation is possible, which is harder when there are many similar plans in the plan library. The next step consists in create distinct top-level nodes (Line 4), which corresponds to agent plans (e.g., plans “p1” and “p2” in Figure 4.5). These nodes are created as simple decomposition type with no features assigned. After that, the algorithm (Lines 5-8) creates their respective branches (this creation is presented in Algorithm 7). Finally, after creates all individual top-level plans, the algorithm selects (Line 9) the ones that will be duplicated (if a duplication percentage has been set). Note, however, that features values of bottom-level nodes from a plan copy are changed in order to distinguish it from the original plan, whenever plans are duplicated.

Algorithm 6 *generateTree*($g, d, b, B, \Phi, F, f, c, \Psi$)

Input: Number of top-level plans g , Depth d , Minimum number of branches b , Maximum number of branches B , Sequential edges Φ , Number of features F , Number of features per node f , Feature status c , Duplication Ψ

Output: Plan Library Λ

```

1:  $\eta_{root} \leftarrow$  create a root node
2:  $\nu_{dup} \leftarrow g \times \Psi$  ▷ number of plans to be duplicated
3:  $\nu_{dist} \leftarrow g - \nu_{dup}$  ▷ number of distinct top plans
4:  $\mathcal{L}_\eta \leftarrow$  create  $\nu_{dist}$  top level nodes
5: for each  $\eta_i \in \mathcal{L}_\eta$  do
6:    $\eta_i \leftarrow$  createBranches( $\emptyset, 1$ )
7:   add decomposition node  $\eta_i$  to  $\eta_{root}$ 
8: end for
9: add  $\nu_{dup}$  duplicated plans to  $\eta_{root}$ 
10: return Plan Library  $\Lambda$  from  $\eta_{root}$ 

```

Algorithm 7 shows the creation of plan-step nodes and their respective branches, receiving as input the subset of features assigned to its parent node ($\mathcal{F} = \emptyset$) and the current depth ($\delta = 1$) in which the node will be created. Some parameters, such as, number of features per node (f), minimum number of branches (b), maximum number of branches, (B) and sequential edges (Φ) are assumed as global (Algorithm 6). The algorithm starts determining the number of branches to be created in the new node

Algorithm 7 *createBranches*(\mathcal{F}_p, δ)

Input: Parent Features \mathcal{F}_p , Current Depth δ **Output:** Branch starting from Node nd

```

1:  $\beta \leftarrow$  number of branches ▷ random number between $b and $B
2:  $\phi \leftarrow$  number of sequential edges ▷  $\Phi$  percent of  $\beta$ 
3:  $\mathcal{F} \leftarrow$  random set of features
4: if  $\mathcal{F}_p \neq \emptyset$  then
5:    $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{F}_p$ 
6: end if
7:  $\eta \leftarrow$  create a new node at depth  $\delta$  with feature set  $\mathcal{F}$ 
8: if  $\delta = \$d$  then
9:   return Node  $\eta$ 
10: end if
11: for  $i = 0$  to  $\phi$  do
12:    $\eta_i \leftarrow$  createBranches( $\emptyset, \delta + 1$ )
13:   add sequential child node  $\eta_i$  to  $\eta$ 
14: end for
15: for  $i = 0$  to  $(\beta - \phi)$  do
16:    $\eta_i \leftarrow$  createBranches( $\mathcal{F}, \delta + 1$ )
17:   add decomposition child node  $\eta_i$  to  $\eta$ 
18: end for
19: return Node  $\eta$ 

```

(Line 1), and how many of them will be set as sequential branches (Line 2). Both information must be defined before the creation of the new node, because they influence how its type is determined. If the number of sequential edges (Φ) is equal to the number of branches (β), then the new node will be created as action node (also referred as “leaf node”), otherwise, it will be created as decomposition node. Next step (Line 3) is to generate the subset of features to be assigned to the new node. This subset, with size defined by $\$f$, is extracted from the feature set ($\$F$) received from the parent node and these features receive values randomly extracted from the previously defined interval $[0, \$c]$. Thus, based on subset of features and information previously determined, the algorithm creates a new node (Line 7). However, before creating branches for a new node is needed to check if current level has reached the expected depth (Line 8). So, if expected depth has been reached, a new node is returned and plan path is completed. Otherwise, the algorithm goes to next step creating a next level in the tree.

The algorithm creates the node sequential branches in Lines (11-14). In the following, it is recursively called until all sequential nodes had been created. Each node created in these recursive calls is added to the list of sequential children of the new node (Line 13). After that, this algorithm uses a similar approach to create the new node decomposition branches (Lines 15-18). The main difference between creating sequential and decomposition branches is the feature distribution among nodes. While creating sequential nodes the feature subset construction is based on the whole feature set ($\$F$),

creating decomposition nodes is based on a feature subset, which eliminates features already used by the parent nodes. As previously explained, the same feature can not be attributed to a node and its decomposition children with different values, so the child node inherits the features, and its respective values, from the parent node (Lines 4-6).

In Figure 4.5, nodes represent plans/goals (first level) and plan steps (second level and below) of the plan library, and edges represent relations between them. The “root” node is not considered a plan, being used only to connect various top plans/goals. Figure 4.5 shows sequential links represented by dashed arrows and decomposition links represented by solid arrows, for instance, there is a decomposition link between p2 and ps2.2, and a sequential link between ps2.2 and ps2.2.1. The top-level plans are p1 and p2 and Figure 4.5 also shows the set of feature status on observable features associated with each plan step.

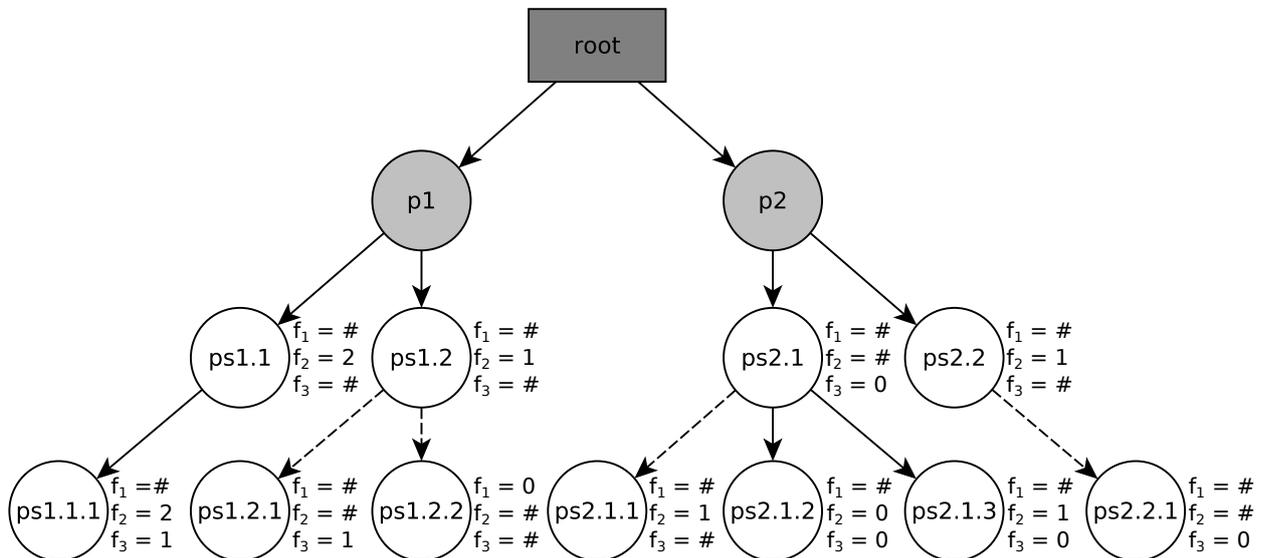


Figure 4.5: Example of a plan library tree created by the *Plan Library Generator* with $\$g=2$, $\$d=3$, $\$b=1$, $\$B=3$, $\Phi=0.5$, $\$F=3$, $\$f=1$, $\$c=2$, and $\Psi=0$.

4.3 Input Set Generation

In this work, we first realise experiments to evaluate SBR performance. In these experiments the observations set used as input for the plan recognition are automatically built using the generated plan library. In the input set, observations are organised in subsets known as “queries”. Each query contains the necessary observations (one or various) for recognition a given plan. The number of observations in a query varies based on the plan structure to be recognised. For example, in Figure 4.5 p1 can be recognised through ps1.1.1 with a single observation, because all path from p1 to ps1.1.1 is com-

posed only by decomposition branches. Otherwise, the path from $p1$ to $ps1.2.2$ has a sequential branch, so we need, at least, two observations owing to the temporal restriction (sequential branch) between $ps1.2$ and $ps1.2.2$. The observation sets creation is conducted automatically, as shown in Algorithm 8 that receives as input a plan library (Λ) and the number of queries (n_{obs}) to be generated.

Algorithm 8 *obsGenerator*(Λ, n_{obs})

Input: Plan Library Λ , Number of Observations set n_{obs}

Output: Observations Sequences List \mathcal{L}_{obs}

```

1:  $\mathcal{L}_{obs} \leftarrow \emptyset$ 
2: for  $i = 0$  to  $n_{obs}$  do
3:    $\eta \leftarrow$  random top-level node
4:    $obs \leftarrow$  getObservations( $\eta$ )           ▷ observations set from a path to perform  $\eta$ 
5:   add  $obs$  to  $\mathcal{L}_{obs}$ 
6: end for
7: return  $\mathcal{L}_{obs}$ 

```

The *obsGenerator*(Λ, n_{obs}) (Algorithm 8) starts randomly selecting a plan from the plan library to be recognised (Line 3). The structure of the chosen plan is traversed and all necessary features (and its values) for the observations generation are got from its plan steps (Algorithm 9). The generation process (Lines 2-6) is repeated until reach the number of expected queries.

Algorithm 9 *getObservations*(η)

Input: Node η

Output: Observations Sequence List obs

```

1:  $obs \leftarrow \emptyset$ 
2:  $\mathcal{L}_\beta \leftarrow$  child nodes of  $\eta$ 
3: if  $\mathcal{L}_\beta = \emptyset$  then
4:    $o \leftarrow$  observation of  $\eta$ 
5:   add  $o$  to  $obs$ 
6:   return  $obs$ 
7: end if
8:  $\eta^* \leftarrow$  random node from  $\mathcal{L}_\beta$ 
9: if  $\eta^*$  is a decomposition child of  $\eta$  then
10:   $o \leftarrow$  getObservations( $\eta^*$ )
11:  add  $o$  to  $obs$ 
12: else
13:   $o \leftarrow$  getPrevObs( $\eta$ )
14:  add  $o$  to  $obs$ 
15:   $o^* \leftarrow$  getObservations( $\eta^*$ )
16:  add  $o^*$  to  $obs$ 
17: end if
18: return  $obs$ 

```

The *getObservations*(η) (Algorithm 9) shows how branches of a given plan are traversed. Initially, each child node of a given node is added to a single list (Line 2).

After, one of these children is randomly selected (Line 8) and if this selected child node is connected by a decomposition branch the method is recursively called using as input this child node (Line 10). For plan paths composed only by decomposition branches (e.g., “p1” to “ps1.1.1”), the method is recursively called until find a leaf-node (e.g., “ps1.1.1”), then all features and its values contained in the leaf node are used to generate an observation. This strategy only is possible because in decomposition relation a child node always inherits all parent features. However, if the selected child is linked by a sequential branch, before the algorithm continues to follow the selected path, it has to generate an observation using the current plan step features, because a sequential branch represents a temporal relation between nodes, where for a child node to be validated, its parent has to be recognised first. For example, given the diagram shown in Figure 4.5 and the assumption that plan steps “ps2.2” and “ps2.2.1” correspond, respectively, to the actions “*at kitchen*” and “*taking medication*”, for one to be able to execute the action “*taking medication*”, one has to execute the action “*at kitchen*” first.

The method to build previous observation is shown in Algorithm 10. In this algorithm, before the execution to follow a sequential path, it tries to build the observation using the current node features. In this situation two cases might occur: First, the current node is a leaf node (i.e., has no decomposition children), thus the observation is built using features and values of the current node (Lines 2-6). Second, the current node is decomposition type, so the strategy used is to follow the decomposition branches of the children nodes until reach a leaf node (Lines 7-9). It is worth noting that the next decomposition node to be consulted is randomly chosen (Line 8).

Algorithm 10 $getPrevObs(\eta)$

Input: Plan Node η

Output: Observations Sequence List obs

- 1: $obs \leftarrow \emptyset$
 - 2: **if** η is a leaf node **then**
 - 3: $o \leftarrow$ observations of η
 - 4: add o to obs
 - 5: **return** obs
 - 6: **end if**
 - 7: $\mathcal{L}_{dec} \leftarrow$ decomposition children nodes of η
 - 8: $\eta^* \leftarrow$ a random node of \mathcal{L}_{dec}
 - 9: $o \leftarrow getPrevObs(\eta^*)$
 - 10: add o to obs
 - 11: **return** obs
-

4.4 Experiments

The experiments aim to demonstrate the usability of the approach presented in this work to generate parameterised test structures, which allows principled performance evaluation for plan recognition approaches. An extensive set of experiments varying a number of parameters was carried out in order to evaluate the performance of the Symbolic Behaviour Recognition (SBR) algorithm described in [AZK05]. SBR is a method for complete and symbolic plan recognition that uses a plan library, which encodes agent’s behaviour in form of plans. It extracts coherent hypotheses from the multi-featured observations sequence using a Feature Decision Tree (FDT) to efficiently match these observations to plan steps in a plan library. A FDT is a decision tree where each node represents an observable feature and each branch represents one possible value of this feature. Determining all matching plans from a set of observations features is efficiently achieved by traversing the FDT top-down until a leaf node is reached. Each leaf node is a pointer to a plan step in the plan library. The *Plan Library Generator* (Section 4.2) builds a plan library based on the given parameters and the *Input Set Generator* (Section 4.3) generates sequences of observations based on this plan library. The algorithms were implemented in Java SDK 1.7 (build 1.7.0_65-b17) and we ran the experiments on a Mac Pro Server (OS X 10.9.4) with two 6-core Intel Xeon (2.4 GHz) CPU, 32 GB of RAM (DDR3 1333MHz), and 2 TB of disk storage. In all experiments we evaluated the performance of SBR varying some parameters and for each of these values we generated a set of 200 random observation based on the given plan library. The average runtime of SBR matching those 200 observations are shown in Figures 4.6, 4.8, 4.10, and 4.12. The FDT training times for each experiment are shown in Figures 4.7, 4.9, 4.11, and 4.13.

The first experiment (Figures 4.6 and 4.7) was performed by varying depth (3 – 10) and sequential edge probability (0 – 1). The other values were fixed as: top level plans = 10; minimum branches = 1; maximum branches = 3; number of features = 10; features per node = 1; feature status = 2 and duplication = 0. Figure 4.6 shows that SBR algorithm is more efficient in domains where plan paths do not have an expected temporal order of execution, *i.e.*, plan paths with few sequential edges. Another important aspect is that the greater the sequence of actions necessary to realise a plan, the greater is the time to recognise it. This is because depth has a strong influence on the size and complexity of the plan library. This experiment was important to demonstrate the possibility of varying temporal structure of the plan library by controlling the number of sequential edges.

The time required to train the FDT is represented in Figure 4.7, where it is possible to observe that time increases according to the plan library depth (the deeper

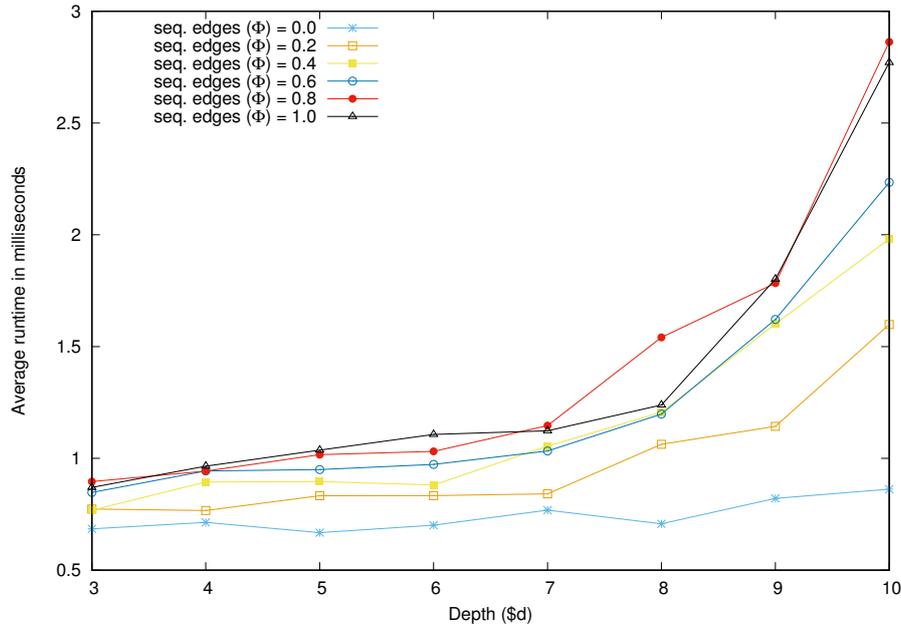


Figure 4.6: Average matching runtime of SBR varying d and Φ .

the plan library, the greater the FDT training time), and the number of sequential edges (the greater the number of sequential edges, the greater the training time).

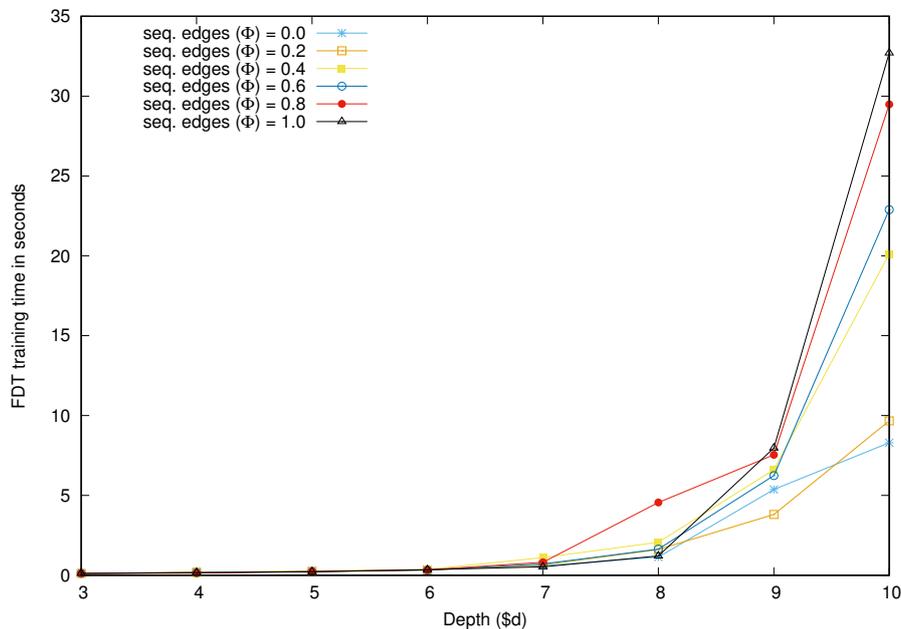


Figure 4.7: FDT training time varying d and Φ .

Figure 4.8 shows the average runtime of SBR varying the number of top-level plans (10 – 100) and number of features per node (1 – 10). The other values were fixed as: depth = 5; minimum branches = 2; maximum branches = 2; number of features = 50; feature status = 2, sequential edges = 0.5 and duplication = 0. In this figure, one

can observe that SBR recognition time tends to increase as the number of top-level plans increases in the domain.

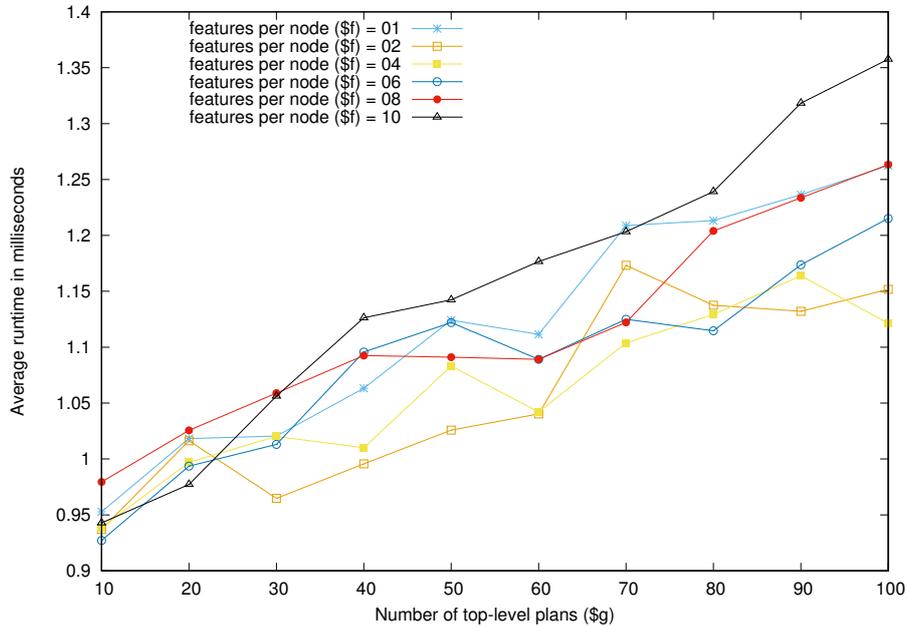


Figure 4.8: Average matching runtime of SBR varying $\$g$ and $\$f$.

Figure 4.9 shows the influence that the number of top-level plans and the number of features per node carried on the FDT training time. We can observe that the increase in number of top-level plans increases the training time. This is contrary to what occurs with the number of features per node, where an increase in the number of features tends to lead to decrease in the time taken for training. This experiment indicates that it is possible to use these algorithms to generate increasingly complex plan-library structures to (stress) test various plan recognition algorithms.

The third experiment (Figures 4.10 and 4.11) was performed by varying the number of branches (1 – 9) and the number of features per node (1 – 10). It is worth noting that for this experiment, in particular, the number of branches represents the exact number of children to each node (*i.e.*, parameters minimum and maximum number of branches assume the same value). The number of top-level plans remains fixed at 10 and the duplication factor was fixed at 0.2, which presents the possibility of increasing ambiguity in plan library provided by plan library generator algorithm. Other parameters assume the same values presented in the second experiment. Figure 4.10 shows that the time for SBR to recognise a plan increases as the number of branches and the number of features per node assumes higher values. This figure show a behaviour that is more regular than that seen in Figure 4.8, which may be due to the increased size and complexity of the plan library; such increase in size and complexity happens more quickly when we change the number of branches than when we change the number of top-level plans.

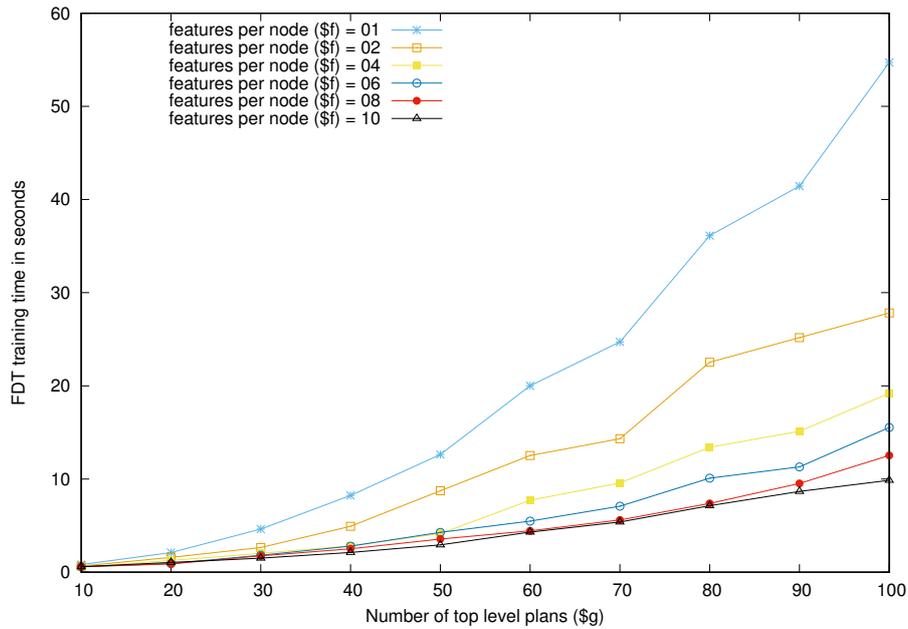


Figure 4.9: FDT training time varying g and f .

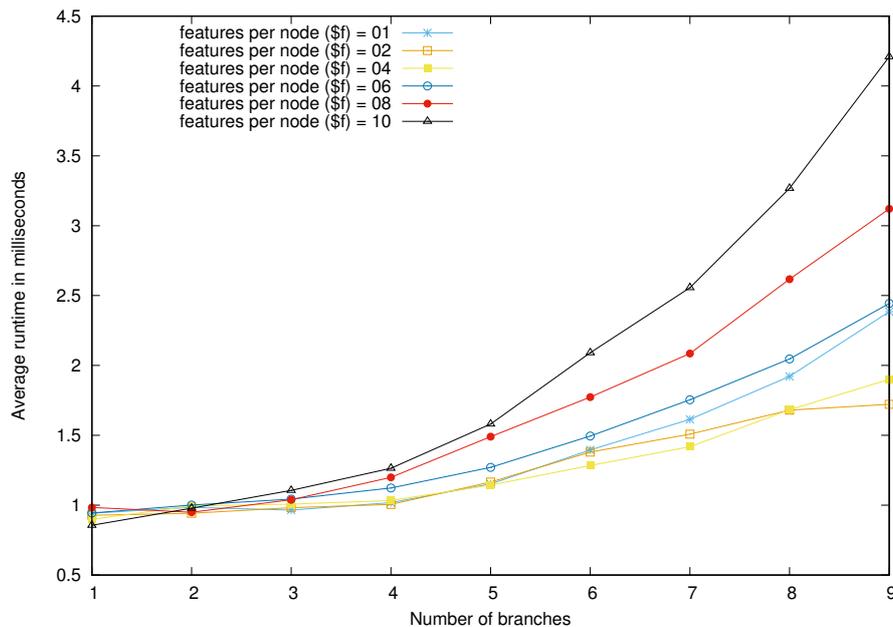


Figure 4.10: Average matching runtime of SBR varying B and f .

The FDT training time presented in Figure 4.11 shows the major influence that the branching factor (number of branches per node) has on the SBR performance, especially for the highest values. Although the time to recognise a plan remains at the scale of milliseconds (see Figure 4.10), the FDT training takes much more time.

Finally, the fourth experiment (Figures 4.12 and 4.13) was performed with the aim of assessing the influence that the number of features per node and the number of values that can be associated to the features (*i.e.*, feature status) have on FDT training

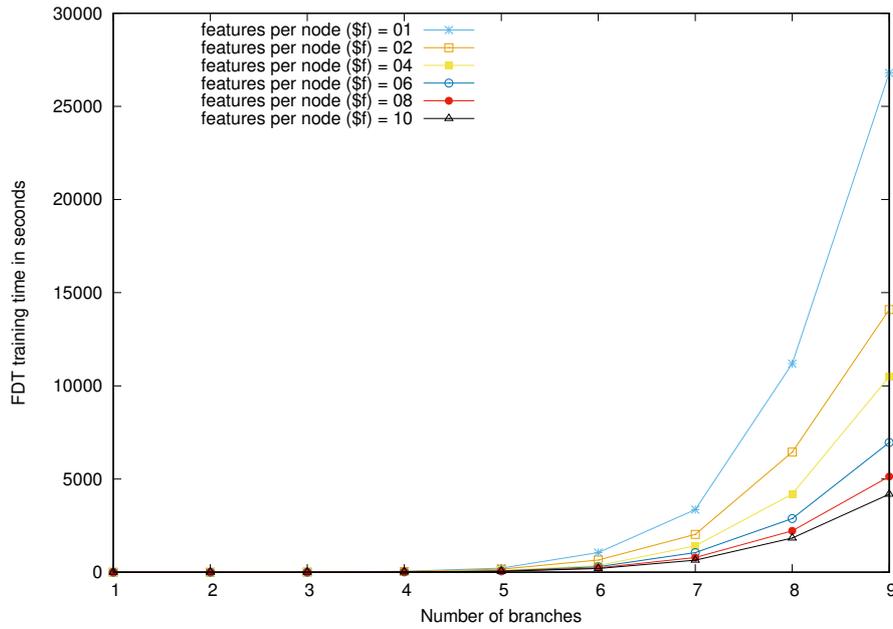


Figure 4.11: FDT training time varying $\$B$ and $\$f$.

time. Figure 4.12 shows average runtime of SBR varying number of features per node (1 – 10) and feature status (1 – 10). The other values were fixed as: top-level plans = 10; depth = 7; minimum branches = 3; maximum branches = 3; number of features = 10; sequential edges = 1 and duplication = 0. This experiment shows that the expected effect of using the FDT is diminished when the number of features per node (plan step) is set at 1, which essentially treats features as atomic.

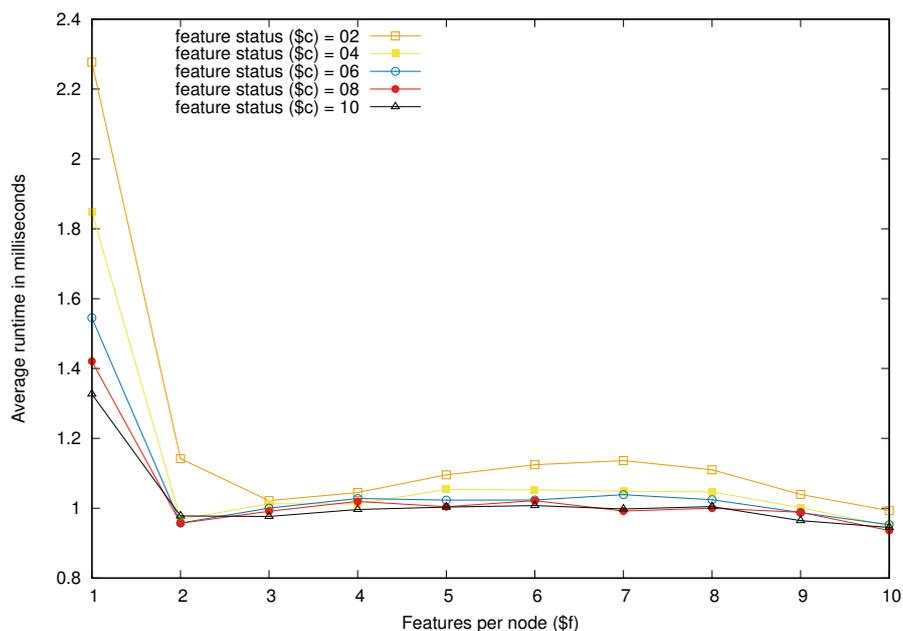


Figure 4.12: Average matching runtime of SBR varying $\$f$ and $\$c$.

In Figure 4.13, we observe that the fewer features per node, the slower the FDT training time (the worst case is when there is only one feature per node). The range of feature status also influences the training time, where we can observe that the greater the relative value of feature status, the smaller the training time.

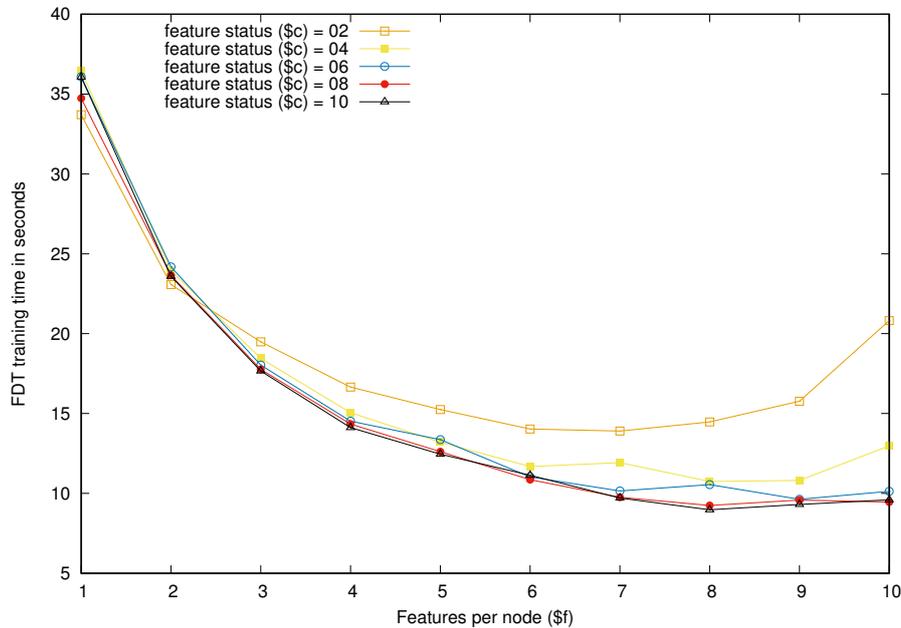


Figure 4.13: FDT training time varying f and c .

Final Remarks

We developed a framework that allows principled performance evaluation for plan recognition algorithms. A plan library generator was created to generate complex structures based on a number of parameters that will determine the complexity of the plan library. Thus, a unique representation of an information domain can be used to compare the efficiency of several plan recognition algorithms. The performance of plan recognition algorithms is directly related to the structure and size of the plan library, as well as the set of observations given to the plan recognition system. The size of a plan library is mainly determined by the number of top-level plans, the interval composed by the number minimum and maximum of branches, and by its depth. On the other hand, the ambiguity of plan library influences the amount of distinct plans that fit a given sequence of observations. In our approach, the amount of ambiguity is determined by: (i) the duplication, which taking larger values implies more duplicated plans, thereby increasing the ambiguity; (ii) the number of features, less features tend to decrease the possibility distinction between plans; (iii) the feature status, which assuming higher

values enables greater distinction among plans that use the same set of features; and (iv) the number of features per node where greater values cause more variety in plans.

Experiments were carried out to assess the effectiveness of this approach by generating several plan libraries with different structures to test the performance of the SBR algorithm. We observed that SBR has a better recognition time performance in domains presenting plan paths with fewer sequential edges and short sequences of actions and it presents worse performance when the number of branches and number of features per node assume higher values. Besides, SBR recognition time and FDT training time tend to increase according with the number of top-level plans in the domain. The FDT training time increases according to plan library depth and number of sequential edges, and in contrast, this value decreases when the number of features per node increases. It is worth noting that the expected effect of using FDT is diminished when the number of features per node is set at 1, which essentially treats features as atomic. Besides, a large range of feature status and a large number of features per node make for the fastest FDT training time. Finally, the experiments demonstrated the capacity of our plan library generator to create increasingly complex and varied structures, as well as the possibility to change the temporal structure and ambiguity of the generated plan library to evaluate plan recognition algorithms. In next chapter, we present the Temporal Multiple-Plan Recognition and the real dataset used to evaluate our approach. We also explain how to extend the Plan Library Generator and the Input Set Generator to create temporal plan libraries and observation set, respectively.

5. TEMPORAL MULTIPLE-PLAN RECOGNITION

This thesis fits into the context of activity and plan recognition, with special attention on increasing interaction between smart devices and people, thanks to hardware and software advances that enable the growth of pervasive computing and the Internet of Things (IoT) [GIMA10], allowing integration of several technologies and communications solutions, stimulating the research in many fields of Artificial Intelligence.

In order to pro-actively help the user without explicit requests, a pervasive computing system must be able to infer what the user is trying to accomplish [Sat01]. The techniques that do this are known as activity and plan recognition. Although both have a similar goal, while activity recognition is the task of recognising the user activity by analysing his/her low-level actions and environment changes, plan recognition is the task of recognising not only an activity but the set of activities (plan) that an observed user is following in order to achieve his/her goal.

Despite some differences, both techniques try to recognise the user intentions by analysing user actions and environment changes, reasoning about how an action from an observed user might contribute to achieve his/her goals. In addition to the part responsible for reasoning, the observer must have a knowledge base that stores user's goals and the ways that they can be achieved, in order to match user's actual behaviour with their intentions. Frequently, user activities and plans are stored in the observer knowledge base and are used for matching the actual user behaviour with some model of expected behaviour.

In this work, we develop a temporal multiple-plan recognition method to deal with ambiguous and concurrent plans, able to handle the several types of devices and the large amount of information they transmit. We extend the SBR algorithm, which supports multi-feature observation and allows multi-valued features, to allow an efficient temporal analysis for all recognition hypotheses consistent with the observations, taking into account concurrent plans and duration constraints.

5.1 Architecture Overview

This section presents an overview of the temporal multiple-plan recognition architecture, shown in Figure 5.1. This model is inspired by ideas presented in Fagundes et al. [FMBV14] but with different goals. While Fagundes focuses on the problem of determining when to interact with the observed agent in order to determine its current plan under execution, we focus in discovering if the observed agent is performing more than a plan at same time, trying to disambiguate the hypothesis set based on information about time of day and duration of each plan execution acquired in previous episodes. Thereby,

the temporal plan recognition is formed by a set of components, operating together, aiming to disambiguate the hypothesis set returned by SBR apart from monitoring temporal errors in plan execution.

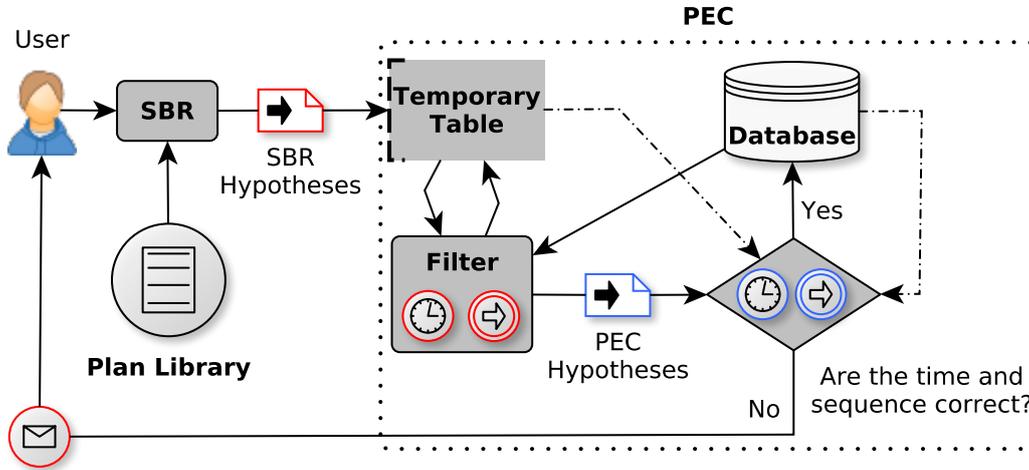


Figure 5.1: Temporal multiple-plan recognition architecture.

The SBR component, presented in Section 2.4 and shown in Figure 5.1, is a symbolic plan recogniser that efficiently matches multi-featured observations to plan steps in a plan library augmented with a feature decision tree. This component receives as input a sequence of observations and matches them to plans stored in the plan library considering all recognition hypotheses that are compatible with the observation history, with no hypotheses ranking. Therefore, SBR can return more than one plan at time, which in most applications means the algorithm could not recognise the correct plan being performed by the user. However, this characteristic can be useful considering that users in real world are multitasking, so the ambiguity present in SBR responses can become an advantage in environments where users perform several tasks concurrently, allowing the algorithm to recognise simultaneous plans being carried out by the user.

In our temporal multiple-plan recognition approach, we call Plan Execution Controller (PEC) the components involved in hypotheses disambiguation process and temporal execution control of a plan. PEC handles hypotheses returned by SBR jointly with current information about these plans stored in *Temporary Table* (Table 5.1 which is hereafter referred to as Υ) and previous information about these stored in *Database* (Table 5.2 which is hereafter referred to as Δ) aiming to monitor plans performed by the user. The *filter* component (present in Figure 5.1) is responsible for disambiguating SBR hypotheses and is extremely important both for plan recognition and for failure prediction. In the recognition process, its task is to improve precision without losing recall, that is, for all candidate plans we must select (filter) only plan(s) actually being performed by the user (precision) and all these plans must be returned as hypothesis

(recall). Like this, in failure prediction its task is to ensure that warning messages will only be sent about plans that are actually being carried out by the user.

Therefore, PEC must be able to identify and check plan steps taking too long as well check plans that have been abandoned before they finished. In other words, it is responsible for checking inconsistencies in plan execution, besides identifying plans that can be executed concurrently. To make this possible, the *Temporary Table* keeps the following information:

- **id** – unique identifier for each plan (or plan step) present in the hypothesis set;
- **label** – a label to define the user activity (*e.g.*, *preparing lunch*);
- **type** – information about the activity (node) position in the sequence of activities performed to achieve a plan/goal, classified as: TOPLEVEL, INTERNAL or EXTERNAL;
- **start** – timestamp where the activity started to be recognised;
- **end** – last timestamp where the activity has been recognised;
- **duration** – number of timestamps elapsed from activity start until its end;
- **path** – sequence of activities performed until the current plan step (only activities associated with the current plan step can belong to this path);
- **hour** – the hour when the activity started being recognised.

Table 5.1: Example of temporary table (Υ)

id	label	type	start	end	duration	path	hour
p1	Preparing Lunch	TOPLEVEL	1	1201	1200	[p1]	12:04:11
ps1.1	Open Refrigerator	INTERNAL	1	61	60	[p1,ps1.1]	12:04:11
p2	Preparing Dinner	TOPLEVEL	1	61	60	[p2]	12:04:11
⋮							
ps1.1.2.1.1	Using Microwave	EXTERNAL	901	1201	300	[p1,ps1.1,... , ps1.1.2.1.1]	12:19:11

It is important to remember that a plan (sequence of activities) is represented by a sequence of nodes in the plan library, where TOPLEVEL nodes denote plans/goals which the agent may try to achieve, so they are top nodes in the plan library. Nodes between top-level and leaf nodes are classified as INTERNAL, representing middle activities performed in a plan execution. Finally, EXTERNAL nodes represent the last activity required to complete a plan successfully; they are leaf nodes in a plan library and used to identify when a plan has finished. Figure 5.2 shows the positions of node types in a plan library.

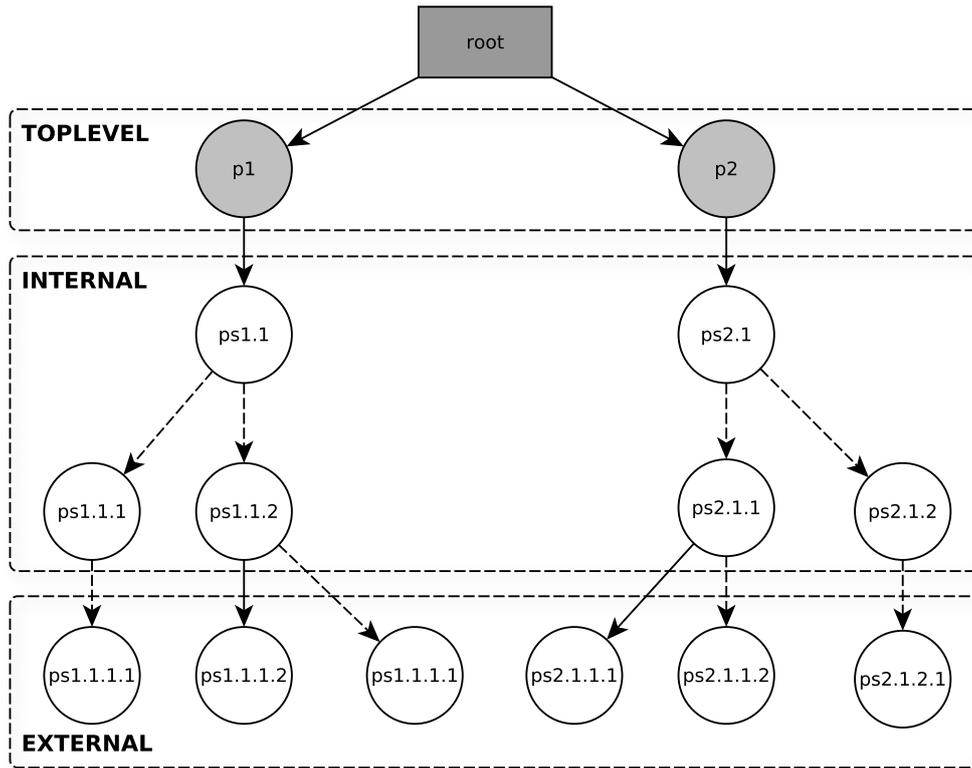


Figure 5.2: Example of node types and their positions in a plan library.

It is worth noting that the *Temporary Table* (Table 5.1) keeps information temporarily, discarding them when the current plan recognition episode ends. However, this information is not ignored, since it is the basis to update the *Database* (Table 5.2) created to store information about plans that have been performed successfully by the agent (information such as plan duration average and how many times a plan was successfully completed). PEC keeps updated both the temporary information present in the temporary table as well as the consistent information stored in the database. The *Temporary Table* is updated at each timestamp, keeping consistent the information of all plans being recognised by SBR and discards the information of plans that are no longer being recognised. The *Database* is update when all the plan recognition process ends (i.e., a plan carried out completely and successfully). So, the information about this plan present in temporary table is used to update the *Database* storing the following information:

- **id** – unique identifier of each activity (node);
- **label** – a label describing the user activity;
- **type** – node type based on position in the activity sequence present in the plan library;

- **tmin** – the smallest number of timestamps required to perform this activity. This value is acquired through many plan executions. Whenever this step presents a duration less than the current *tmin* we update this information with the new duration;
- **tmax** – the longest duration to finish this plan step among all plan executions. We update this value whenever this step presents a duration higher than the current *tmax*;
- **tavg** – average duration (in timestamps) to finish this activity;
- **nupd** – number of times that the plan information has been updated;
- **hour** – average hour at which this activity is usually performed. When the activity has no specific time of day to take place we tagged this as UNDEFINED.

Table 5.2: Example of database (Δ)

id	label	type	tmin	tmax	tavg	nupd	hour
p ₁	Preparing Lunch	TOPLEVEL	1200	1500	1350	2	11:59:06
ps _{1.1}	Using Refrigerator	INTERNAL	60	60	60	1	12:04:11
ps _{1.2}	Using Cabinet	INTERNAL	90	90	90	1	11:54:01
p ₂	Preparing Dinner	TOPLEVEL	1800	1800	1800	1	20:05:00
⋮							
ps _{1.1.2.1.1.2}	Using Microwave	EXTERNAL	300	300	300	1	12:24:06

The hour is not an information previously informed to the temporal plan recogniser, that is, the system must learn and update this value each time a plan is carried out by the user. So, plans that usually are performed at same hour possibly will have an average hour defined, otherwise they will be tagged as UNDEFINED. The first time a plan is performed we assume the current time as the average hour. In next episodes, the hour is updated according to the following Equation 5.1

$$hour = \begin{cases} \frac{(hour \times nupd) + obs_hour}{nupd + 1} & \text{if } |hour - obs_hour| \leq \varphi \\ \text{UNDEFINED} & \text{Otherwise} \end{cases} \quad (5.1)$$

where the variable *obs_hour* represents the current hour when the plan started execution, and we use constant φ as the maximum time variation range allowed between the times when the plan execution occurred to consider that the plan has a definite hour to take place. In other words, we calculate the absolute difference between two time values, the average hour *hour* and the current hour *obs_hour* (when the plan started being

recognised). If this difference is less than or equal to φ we update the average hour, otherwise, we consider the plan has no defined hour (tagging it as UNDEFINED).

5.2 Plan Execution Controller

The PEC components extending the SBR algorithm aim to disambiguate the hypotheses set returned by SBR when it presents more than one element. These components are also able to determine when an agent is performing more than one plan at the same time, besides monitoring the temporal order of the activities (i.e., the activity sequence) and checking their time and duration. Information about the plans present in the hypotheses set are acquired from the plan library, such as *id*, *label*, *type*, and *path*. The node *type* is defined according to its position in the plan tree. The *path* information contains all traversed nodes from the top to the current node. These values, as well as *start*, *end*, *duration*, and *hour* are updated at runtime, kept in temporary table Υ , and discarded when the current plan recognition episode ends.

To develop a mechanism able to control the time execution for each activity performed by the user, it was necessary to make some assumptions on the way observations are made. We assume that observations are made at fixed intervals of time, i.e, in our experiments, for example, a new observation is made every second. This fixed interval between each observation allows us to accurately rate and store the time sequence for each activity being performed, thereby enabling us to check (at runtime) temporal inconsistencies in a plan execution.

Algorithm *checkHypothesis* (see Algorithm 11) is responsible for checking each plan node information stored in the temporary table Υ (Line 1), i.e., all information about plans and plan steps recently recognised by SBR. Among these nodes, we should pay attention to the ones no longer being recognised, because it means their execution may have finished. Then, we must check if they show some execution time inconsistency (e.g., a plan was performed faster than its minimum expected duration) or sequence error (e.g., a plan stopped being recognised without reaching an EXTERNAL node). These checks are performed by *checkPlan* (Line 3) described below in Algorithm 12. On the other hand, a node still being recognised must have its time duration checked to ensure that its plan step is not taking too long (Line 5).

It is worth noting that the value of t represents the current recognition timestamp and the equation $t - \gamma$ (assuming $\gamma \geq 1$) is the last timestamp where the node η_i was returned as hypothesis in the plan recognition episode. We use γ to define how many recognition episodes are required in order to assert that a particular plan has finished or has been abandoned. In this work, we implement a SBR variant, presented in Avrahami-Zilberbrand [AZKZ05], allowing the recognition of interleaved plans (i.e., an agent may

interrupt a plan to execute another, only returning to the remaining plan steps of the first plan later). To handle interleaving, we limit this time by using γ as a parameter to determine how many time units (timestamps) can pass between the time when a plan step was interrupted and the time its execution is resumed.

Algorithm 11 *checkHypothesis*(Δ, Υ, t)

Input: Database Δ , Temporary Table Υ , Timestamp t

```

1: for each  $\eta_i \in \Upsilon$  do
2:   if  $\eta_i$  has finished at  $(t - \gamma)$  then
3:     checkPlan( $\eta_i, \Delta, \Upsilon, t$ )
4:   else
5:     check if  $\eta_i$  is not taking too long
6:   end if
7: end for

```

The *checkPlan* method (see Algorithm 12) is responsible for checking each node η according to its position in a plan sequence, and informing any anomaly in plan execution to the user. This method receives as input a node η , the database Δ , the temporary table Υ and the current timestamp t . First, this algorithm finds the TOPLEVEL node η^* associated with the current node η (i.e., the plan which the current plan step is part). Knowing the top node associated with the current node is important because before performing any error analysis we must ensure that this top node meets the constraints we use to check if the plan is actually being performed (Line 2), otherwise we must discard this hypothesis.

In Algorithm 12, we assume that η has finished in a previous timestamp. In this case, supposing η as a node (plan step) linked to a plan η^* that was actually being performed by the user and correctly recognised by the temporal plan recognition, we must check its execution time according to node type (Line 4). The case where η is a TOPLEVEL node (Line 5) indicates that a plan has finished or has been abandoned. Thus, we need to find an EXTERNAL node η_ε associated with η (Line 6), that was performed by the user, to indicate that all plan steps were carried out. If there is an EXTERNAL node and it was performed taking more time than its minimum duration (Line 7), we may consider that the plan was performed successfully and we therefore save/update all node information associated with this plan in the database Δ (Line 8). Otherwise, in case ($\eta_\varepsilon = null$), this plan may have been abandoned by the user or its last step has failed (e.g., the last step was performed too fast compared with previous executions, so it probably was not fully completed). In both cases, a message must be sent to the user (Line 10).

The case where η is an INTERNAL node (Line 12) means that a middle plan step has finished or has been abandoned, so we must check if this node was performed taking more time than its minimum duration (Line 13). In the affirmative case, means that η was completed successfully and the plan went forward to next step but it was not completed

yet. Otherwise, η took less time than usual, so a message must be sent to the user warning this possible plan step failure (Line 16).

Finally, when η is an EXTERNAL node (Line 18), this means that a leaf plan step has finished or has been abandoned. In this work, we consider EXTERNAL nodes as the last step (activity) which must be completed for a plan to have finished successfully. Thus, similarly to the INTERNAL case, we must check if the EXTERNAL node was performed taking more time than its minimum duration (Line 19). A correct node execution indicates that the plan has finished successfully, otherwise a message must be sent to the user, because the plan may have been abandoned or failed in this last step (Line 22).

Algorithm 12 *checkPlan*($\eta, \Delta, \Upsilon, t$)

Input: Node η , Database Δ , Temporary Table Υ , Timestamp t

```

1:  $\eta^* \leftarrow$  TOPLEVEL node present in the path of  $\eta$ 
2:  $\mathcal{L}_\eta \leftarrow filterHypothesis(\{\eta^*\}, \Delta, \Upsilon, t)$ 
3: if  $\mathcal{L}_\eta \neq \emptyset$  then
4:   switch type( $\eta$ ) do
5:     case TOPLEVEL
6:        $\eta_\varepsilon \leftarrow$  EXTERNAL node of  $\eta$  present in the  $\Upsilon$ 
7:       if ( $\eta_\varepsilon \neq null$ )  $\wedge$  checkTMin( $\eta_\varepsilon, \Delta, \Upsilon$ ) then
8:         savePath( $\eta, \eta_\varepsilon, \Delta, \Upsilon$ )
9:       else
10:        Send message to the user  $\triangleright \eta$  might be incomplete
11:      end if
12:     case INTERNAL
13:       if checkTMin( $\eta, \Delta, \Upsilon$ ) then
14:         plan not completed yet
15:       else
16:        Send message to the user  $\triangleright \eta$  may not have finished successfully
17:      end if
18:     case EXTERNAL
19:       if checkTMin( $\eta, \Delta, \Upsilon$ ) then
20:         plan completed successfully
21:       else
22:        Send message to the user  $\triangleright \eta$  may not have finished successfully
23:      end if
24:   end if

```

When a plan has carried out as expected, that is, each activity occurred in the right sequence within the given time intervals, the whole path from top-level plan to leaf node must be saved/updated in the database Δ . For this purpose, the method *savePath* (Algorithm 13) was created, receiving as input a TOPLEVEL node η , an EXTERNAL node η_ε associated with η , the database Δ and the temporary table Υ . This method finds out every node η_i belonging to plan path from η through to η_ε , saving their execution data in Δ and deleting them from Υ (Lines 2-6).

Algorithm 13 *savePath*($\eta, \eta_\varepsilon, \Delta, \Upsilon$)

Input: Node η , External Node η_ε , Database Δ , Temporary Table Υ

```

1: if  $type(\eta) = \text{TOPLEVEL}$  then
2:   for each  $\eta_i \in \Upsilon$  do
3:     if ( $\eta_i$  is a node in the path from  $\eta$  to  $\eta_\varepsilon$ ) then
4:       update  $\eta_i$  in the  $\Delta$ 
5:       remove  $\eta_i$  from the  $\Upsilon$ 
6:     end if
7:   end for
8: end if

```

A hard problem that plan recognition algorithms faces is how to disambiguate multiple hypotheses (plans) that are consistent with a sequence of observed user activities. Therefore, recognition algorithms must be able to overcome the ambiguity inherent to real environments, due to the amount of sensor activities that several activities have in common. The SBR algorithm maintains all candidate plans (hypotheses) consistent with the observations, even though there is only a single hypothesis that is the correct one. Unfortunately, this number of possible hypotheses can be very large in practice, especially in environments with few features and many plans available, as we found in our experiments with a dataset based on real world data. In practice, a plan under execution is recognised with precision only when all hypotheses but one have been ruled out. However, returning only one hypothesis can be hard to achieve since SBR may recognise multiple hypotheses for a long time due to ambiguity in the plan library.

The *filterHypotheses* method (see Algorithm 14) faces the problem of disambiguating many possible hypotheses that are all consistent with the actions of the observed agent. This method aims to reduce the set of possible hypotheses without removing the correct ones. This algorithm receives as input a hypothesis list \mathcal{L}_η (candidate plans), the database Δ , the temporary table Υ , and the timestamp t . Basically, this method checks all candidate plans (Lines 2-19) and tries to remove all plans incorrectly recognised by the plan recognition based on the extra information available to it. Only TOPLEVEL nodes in the candidate plans are checked (Line 3); other node types are ignored because, necessarily, they will be part of some path associated with one of these top plans, being kept or removed from the hypotheses set according to the evaluation of their associated top plan. After certifying η_i as TOPLEVEL node, we try to find an EXTERNAL node η^* associated with this top node (Line 4) in the temporary table Υ . Finding this node is important since it means the plan was performed until the last step, denoting a great chance that this plan really is being carried out by the user.

Assuming that an EXTERNAL node was found (Line 5), we must check if the plan is being executed at a suitable time of day. This verification is made by the *isOnTime* method (see Algorithm 15), receiving the top plan η_i , the database Δ , and a range value φ as input. This method checks the expected execution time h_1 acquired in previous exe-

Algorithm 14 *filterHypotheses*($\mathcal{L}_\eta, \Delta, \Upsilon, t$)**Input:** Node List \mathcal{L}_η , Database Δ , Temporary Table Υ , Timestamp t **Output:** Node List $\mathcal{L}_{\eta_{out}}$

```

1:  $\mathcal{L}_{\eta_{out}} \leftarrow \emptyset$ 
2: for each  $\eta_i \in \mathcal{L}_\eta$  do
3:   if  $\eta_i$  is a TOPLEVEL node then
4:      $\eta^* \leftarrow$  EXTERNAL node of  $\eta_i$  present in the  $\Upsilon$ 
5:     if  $\eta^* \neq null$  then  $\triangleright \eta_i$  has finished at an EXTERNAL node
6:       if ( $isOnTime(\eta_i, \Delta, \varphi) \wedge (pathLength(\eta_i, \Upsilon) > \delta) \vee isUnique(\eta_i, \Upsilon)$ ) then
7:          $\mathcal{L}_{\eta_{out}} \leftarrow \mathcal{L}_{\eta_{out}} \cup \{\eta_i\}$ 
8:       end if
9:     else
10:      if  $isLongestDuration(\eta_i, \Upsilon) \vee isLongestPath(\eta_i, \Upsilon)$ 
11:         $\hookrightarrow \vee isUniqueOnTime(\eta_i, \Delta, \Upsilon, \varphi)$  then
12:           $\mathcal{L}_{\eta_{out}} \leftarrow \mathcal{L}_{\eta_{out}} \cup \{\eta_i\}$ 
13:        else
14:          if  $isOnTime(\eta_i, \Delta, \varphi) \wedge (pathLength(\eta_i, \Upsilon) > \delta)$ 
15:             $\hookrightarrow \wedge (isLongerPath(\eta_i, \Upsilon) \vee isLongerDuration(\eta_i, \Upsilon))$  then
16:               $\mathcal{L}_{\eta_{out}} \leftarrow \mathcal{L}_{\eta_{out}} \cup \{\eta_i\}$ 
17:            end if
18:          end if
19:        end if

```

cutions and stored in database Δ , against the current time h_2 . If the difference between these times is less than or equal to the range φ , this means the plan is being performed at a plausible time (i.e., given the available information it is plausible to assume that the user might indeed be executing that plan). The value of φ is important because, in the real world, we can find plans usually performed at a determined period of the day (e.g., making breakfast). These plans do not have a specific hour, but a range of hours in which they usually might occur. Based on this, we store the average hour and together with φ value check if a plan was being performed at the expected time. Plans with no specific execution time (i.e., plans that can be performed at any time of the day) have their time set as UNDEFINED. It is important to note that, for these plans, *isOnTime* will always return true.

Algorithm 15 *isOnTime*(η, Δ, φ)**Input:** Node η , Database Δ , Range φ **Output:** A Boolean value β

```

1:  $\beta \leftarrow false$ 
2:  $h_1 \leftarrow$  expected execution horary for  $\eta$  stored in  $\Delta$ 
3:  $h_2 \leftarrow$  current execution horary for  $\eta$ 
4: if ( $|h_1 - h_2| \leq \varphi$ ) then
5:    $\beta \leftarrow true$ 
6: end if

```

In Algorithm 14, it is further checked the plan path length (i.e., how many plan steps were completed successfully so far). This value is obtained through the *pathLength* method, which traverses the temporary table Υ counting nodes (already completed) associated with the top-level plan η . The δ value represents the minimum number of completed plan steps necessary to set a candidate plan as a correct hypothesis. This constraint is relevant mainly in domains where many plans have UNDEFINED execution times, so another way to disambiguate a hypothesis set may be useful. It is worth noting that *filterHypotheses* tries to disambiguate the candidate plans, but when the recognition process returns only one plan we assume this hypothesis as correct, independently of time and path length. This assumption is checked by *isUnique* method that traverses Υ checking if η is the unique top-level plan currently being recognised.

When an EXTERNAL node associated with the hypothesis η cannot be found, it means that this plan has failed or has been abandoned. However, before notifying the user about this failure or abandonment, we must ensure the plan was actually being performed, i.e., it is not just a false positive. For this purpose several methods are created to check candidate plans, trying to find some specific property related to each hypothesis. For instance, the method *isLongestDuration* checks if η is the hypothesis with the longest duration in Υ (i.e., checks if η is the node for the plan that has been recognised for the longest time, compared to other plans in the hypothesis set). On the other hand, the method *isLongestPath* checks if η is the hypothesis with the most steps completed so far. There is also one variation for each of them (both *isLongerDuration* and *isLongerPath*) which take into account only hypotheses that started being recognised at the same node timestamp or later. This is important because it enables the algorithm to recognise several parallel plans regardless the time each one has started to be performed. We can also check if η is the unique top-level plan at the correct time, returned in the recognition episode, through the method *isUniqueOnTime*.

These methods allow multiple plan recognition since they do not focus only on a single characteristic for plan disambiguation, allowing a more complete, controlled, and detailed analysis of each plan execution. In addition to that, they help decrease ambiguity inherent in the recognition process by eliminating hypotheses that do not fit known domain characteristics.

5.3 Datasets

One of the first publicly available corpora for training and testing plan recognition systems was The Monroe Corpus created by Blaylock and Allen [BA05] where they propose the use of an AI planner and Monte Carlo simulation to stochastically generate synthetic plan corpora [BA05]. This method provides a corpus accurately labelled with goal and hierarchical plan structure and can be used for any domain. However,

this method does not create time constraints for each plan execution nor concurrent or interleaving plans. The increase in studies by several research groups working on smart homes ¹ has led to Activities of Daily Living (ADL) datasets being made publicly available [FHB⁺08, WKL⁺08]. Some relevant smart environment projects with publicly available datasets include the CASAS testbed at Washington State [RC09], Georgia Tech’s Aware Home [AM05] and MIT’s House_n [TIL04]. However, there are still very few publicly available datasets for plan recognition reporting the execution time for each activity, and providing higher-level information besides the sensors’ state. Furthermore, most datasets decompose human activities as a sequence of binary sensor activations and restrict user interaction with a single sensor at time.

Among the datasets surveyed, the files used by Tapia [Mun03] seem to be particularly interesting for this work because it contains information acquired from sensors in a simple way (i.e., the data is stored in text format and coma separated values), keeping the activity duration carried out by the subject through start time and end time for each activity. In addition, this work presents an analysis of all sequence of activities classifying them into categories and subcategories (high-level representation), i.e., modelling the dependency between the activities and the corresponding sensors. This dataset was created by several reed switch sensors [TMIL04] that were installed in two single-person apartments to gather data about human activity for 14 days. Everyday objects (drawers, refrigerators, containers, etc.) received these sensors to record activation/deactivation events from the person performing everyday activities. Both subjects lived alone in one-bedroom apartments, the first person was a professional 30-year-old woman who spent free time at home, and the second was an 80-year-old woman who spent most of her time at home. During the study, the subjects used the context-aware Experience Sampling Method (ESM) to create a detailed record of their activities. Table 5.3 shows the number of training examples per activity generated by indirect observation, where column “Subject 1” represents the number of activities performed by 30-year-old woman and column “Subject 2” refers to 80-year-old woman, respectively. Table 5.4 is part of Table 5.3 where activities with less than six examples were eliminated.

The text dataset ² used by Tapia [Mun03] contains three files (sensor.csv, activities.csv, and activities_data.csv) which contain the data of a specific subject. The file sensor.csv contains the sensor information in the following format:

Sensor_Id,Location,Object

Table 5.5 shows part of the content stored in the sensors.csv file (e.g., 100,Bathroom,Toilet Flush 101,Bathroom,Light switch 104,Foyer,Light switch,...).

¹Dwellings equipped with several sensors and actuators able to monitor and adjust home control system settings in order to provide greater comfort and safety to the inhabitants.

²<http://courses.media.mit.edu/2004fall/mas622j/04.projects/home/>

Table 5.3: Complete table of number of examples per activity generated by indirect observation [Mun03]

Activity	Subject 1	Subject 2
Bathing	18	3
Toileting	84	37
Going out to work,	12	-
Preparing lunch	15	21
Preparing dinner	10	14
Preparing breakfast	14	17
Dressing	24	5
Grooming	37	2
Preparing a snack	15	16
Preparing a beverage	15	1
Washing dishes	8	20
Doing laundry	19	-
Cleaning	9	3
Putting away dishes	2	3
Washing hands	1	-
Putting away groceries	2	1
Other	1	2
Watching TV	3	15
Going out for shopping	2	3
Going out for entertainment	1	1
Lawnwork	1	1
Putting away laundry	2	1
Taking medication	-	14
Listening to music	-	17
Talking on telephone	-	4
Working at computer	-	5
Home education	-	2

Table 5.4: Partial table of number of examples per activity generated by indirect observation [Mun03]

Activity	Subject 1	Subject 2
Preparing dinner	10	14
Preparing lunch	15	21
Listening to music	-	17
Taking medication	-	14
Toileting	84	37
Preparing breakfast	14	17
Washing dishes	8	20
Preparing a snack	15	16
Watching TV	-	15
Bathing	18	-
Going out to work	12	-
Dressing	24	-
Grooming	37	-
Preparing a beverage	15	-
Doing laundry	19	-
Cleaning	9	-

Table 5.5: Sample of the sensors.csv file contents

Sensor_Id	Location	Object
100	Bathroom	Toilet Flush
101	Bathroom	Light switch
104	Foyer	Light switch
105	Kitchen	Light switch
106	Kitchen	Burner
...

The file activities.csv contains all the activities analysed, represented as shown in sequence below:

Heading,Category,Subcategory,Code

Table 5.6 shows part of the content stored in the activities.csv file.

Table 5.6: Sample of the activities.csv file contents

Heading	Category	Subcategory	Code
Employment related	Employment work at home	Work at home	1
Employment related	Travel employment	Going out to work	5
Personal needs	Eating	Eating	10
Personal needs	Personal hygiene	Toileting	15
Personal needs	Personal hygiene	Bathing	20
...

The file activities_data.csv contains the data of the activities as shown in Table 5.7 which represents the organisation of that information. An example showing part of this file is presented in Table 5.8.

Table 5.7: Data organisation in activities_data.csv

activity_label	date	start_time	end_time
sensor1_id	sensor2_id	...	
sensor1_object	sensor2_object	...	
sensor1_activation_time	sensor2_activation_time	...	
sensor1_deactivation_time	sensor2_deactivation_time	...	

5.4 Temporal Plan Library Generator

The temporal plan library generator is an extension of *Plan Library Generator* presented in Section 4.2 aiming to add for each plan step present in the plan library information about minimum ($\$t$) and maximum ($\T) number of timestamps required to perform the activity. Table 5.9 shown all parameters available to generate a plan library

Table 5.8: Part of the data in activities_data.csv file

Toileting 67	3/27/2003 100	6:42:4 101	6:45:55 57	58 Medicine cabinet	67 Cabinet	82 Drawer	71 Drawer
Cabinet 6:43:40	Toilet Flush 6:44:6	Light switch 6:44:20	Medicine cabinet 6:44:35	Medicine cabinet 6:44:36	Cabinet 6:44:49	Drawer 6:45:45	Drawer 6:46:12
6:43:43	7:12:41	7:46:34	6:44:48	6:44:48	6:44:56	6:45:48	6:46:20
Preparing breakfast 80	3/27/2003 143	6:50:8 55	6:57:56 54				
Cabinet 6:51:43	Microwave 6:54:9	Cabinet 6:54:16	Door 6:57:5				
6:51:46	13:7:43	6:54:19	6:57:8				
Going out to work 140	3/27/2003	7:47:41	7:51:25				
Door 7:48:49							
7:48:53							
Preparing breakfast 84	3/27/2003 72	7:38:29 73	7:41:49 70	135 Drawer	91 Refrigerator	73 Cabinet	58 Medicine cabinet
Drawer 7:38:48	Cabinet 7:39:7	Cabinet 7:39:14	Dishwasher 7:40:32	7:40:46	7:41:8	7:41:36	7:42:38
7:38:51	7:40:27	7:39:17	7:40:34	7:40:49	7:41:16	7:41:37	7:43:52
Dressing 57	3/27/2003 100	7:34:56 82	7:37:50 75	82 Drawer	75 Drawer	84 Drawer	
Medicine cabinet 7:34:5	Toilet Flush 7:34:47	Drawer 7:35:27	Drawer 7:35:50	7:36:23	7:37:18	7:38:48	
7:42:38	12:26:27	7:35:34	7:35:56	7:36:27	7:37:43	7:38:51	
Grooming 73	3/27/2003 58	7:42:26 88	7:44:31 68	57 Medicine cabinet	57 Medicine cabinet		
Cabinet 7:41:36	Medicine cabinet 7:42:38	Sink faucet - cold 7:43:22	Sink faucet - hot 7:43:23	7:43:52	7:44:10		
7:41:37	7:43:52	7:43:48	7:43:48	7:44:6	12:26:50		
Preparing a beverage 140	3/27/2003 92	16:27:17 91	16:48:55 80	91 Refrigerator			
Door 16:27:59	Light switch 16:28:23	Refrigerator 16:35:40	Cabinet 16:35:49	16:36:5			
16:28:3	18:44:14	16:35:45	16:35:50	16:36:9			

more minimum ($\$t$) and maximum ($\T) number of timestamps (listed in grey highlighted lines) which make possible to join temporal aspects to plans in the library.

Table 5.9: Parameters available to generate the temporal plan library

Flag	Symbol	Default	Constraints	Description
-g	$\$g$	$\$g = 3$	$\$g \geq 1$	number of top-level plans
-d	$\$d$	$\$d = 4$	$\$d \geq 1$	depth of the plan trees
-b	$\$b$	$\$b = 1$	$1 \leq \$b \leq \B	minimum number of branches
-B	$\$B$	$\$B = 3$	$\$b \leq \B	maximum number of branches
-f	$\$f$	$\$f = 1$	$\$f \geq 1$	number of features per node
-c	$\$c$	$\$c = 2$	$\$c \geq 1$	number of values
-F	$\$F$	$\$F = 10$	$\$F \geq (\$d \times \$f)$	available values to the features
-s	Φ	$\Phi = 0.8$	$0 \leq \Phi \leq 1$	sequential edges probability
-D	Ψ	$\Psi = 0$	$0 \leq \Psi \leq 1$	duplication probability
-t	$\$t$	$\$t = 1$	$\$t \geq 1$	minimum number of observations
-T	$\$T$	$\$T = 1$	$\$T \geq \t	maximum number of observations
-p	path	<i>plan_library.xml</i>	$\langle String \rangle$	output file path

We improved the methods of plan library generator to support temporal features. The main changes in this generator are listed in grey highlighted lines in algorithms presented in this section. In method *generateTree* (Algorithm 16) we added new parameters as input ($\$t$ and $\$T$) used in the method *createBranches* (Algorithm 17) responsible for creating branches for the top-level plan (Line 6).

Algorithm 16 *generateTree*($\$g, \$d, \$b, \$B, \Phi, \$F, \$f, \$c, \Psi, \$t, \$T$)

Input: Number of top-level plans $\$g$, Depth $\$d$, Minimum number of branches $\$b$, Maximum number of branches $\$B$, Sequential edges Φ , Number of features $\$F$, Number of features per node $\$f$, Feature status $\$c$, Duplication Ψ , Minimum Timestamp $\$t$, Maximum Timestamp $\$T$

Output: Plan Library Λ

- 1: $\eta_{root} \leftarrow$ create a root node
 - 2: $\nu_{dup} \leftarrow \$g \times \Psi$ ▷ number of plans to be duplicated
 - 3: $\nu_{dist} \leftarrow \$g - \nu_{dup}$ ▷ number of distinct top plans
 - 4: $\mathcal{L}_\eta \leftarrow$ create ν_{dist} top level nodes
 - 5: **for each** $\eta_i \in \mathcal{L}_\eta$ **do**
 - 6: $\eta_i \leftarrow$ *createBranches*($\emptyset, 1, \$t, \T)
 - 7: add decomposition node η_i to η_{root}
 - 8: **end for**
 - 9: add ν_{dup} duplicated plans to η_{root}
 - 10: **return** Plan Library Λ from η_{root}
-

Algorithm 17 creates plan-step nodes and their respective branches, receiving as input parameters presented in Algorithm 7 more two new values (t_{min} and t_{max}) representing, respectively, the minimum and maximum number of timestamps necessary to perform a plan step. These parameters are used by the method responsible for creating a new node (Line 10). It is worth noting that t_{min} and t_{max} are global parameters,

Algorithm 17 $createBranches(\mathcal{F}_p, \delta, t_{min}, t_{max})$ **Input:** Parent Features \mathcal{F}_p , Current Depth δ , Min Timestamp t_{min} , Max Timestamp t_{max} **Output:** Branch starting from Node nd

```

1:  $\beta \leftarrow$  number of branches  $\triangleright$  random number between $b and $B
2:  $\phi \leftarrow$  number of sequential edges  $\triangleright \Phi$  percent of  $\beta$ 
3:  $\mathcal{F} \leftarrow$  random set of features  $\triangleright \mathcal{F} \cap \mathcal{F}_p = \emptyset$  and  $|\mathcal{F}| \geq $f$ 
4: if  $\mathcal{F}_p \neq \emptyset$  then
5:    $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{F}_p$ 
6:    $\mathcal{T} \leftarrow [t_{min}; t_{max}]$ 
7: else
8:    $\mathcal{T} \leftarrow$  an integer sub-interval of  $[t_{min}; t_{max}]$  chosen randomly
9: end if
10:  $\eta \leftarrow$  new node at depth  $\delta$ , feature set  $\mathcal{F}$  and timestamp interval  $\mathcal{T}$ 
11: if  $\delta = $d$  then
12:   return Node  $\eta$ 
13: end if
14: for  $i = 0$  to  $\phi$  do
15:    $\eta_i \leftarrow createBranches(\emptyset, \delta + 1, $t, $T)$ 
16:   add sequential child node  $\eta_i$  to  $\eta$ 
17: end for
18: for  $i = 0$  to  $(\beta - \phi)$  do
19:    $\eta_i \leftarrow createBranches(\mathcal{F}, \delta + 1, \min(\mathcal{T}), \max(\mathcal{T}))$ 
20:   add decomposition child node  $\eta_i$  to  $\eta$ 
21: end for
22: return Node  $\eta$ 

```

thus when the algorithm creates a new node (plan step), values between the interval $[t_{min}, t_{max}]$ are randomly generated to determine the minimum and maximum timestamps specific to the new node. So, each plan step in the plan library tends to present different intervals of timestamps execution, however, decomposition nodes inherit the interval of timestamps of their parent (Line 6). The algorithm also creates all sequential and decomposition branches recursively passing t_{min} and t_{max} values (Lines 14 and 21).

5.5 Temporal Input Set Generation

The algorithms developed to represent the architecture presented in Section 5.1 demand a dataset containing information about task execution time, temporal relation between nodes, observations with multiples features as well as making possible features multivalued. Public datasets with all these peculiarities are extremely hard to find, so we developed a temporal plan library generator (Section 5.4) and we improved the input set generator (Section 4.3) to support temporal plan libraries to build observations set.

In experiments, we must get the same features set repeatedly to simulate plan step duration. This information is present in temporal plan library, so we adapted the

method presented in Algorithm 18 to get observations from a library, taking into account information about plan step duration (t_{min} and t_{max}). Thus, the algorithm generates a random number based on these two values (Lines 5 and 17) defining how many times the features will be repeated in the observations set (Lines 6-8 and 18-20) to simulate the activity duration.

Algorithm 18 *getObservations(η)*

Input: Node η

Output: Observations Sequence List obs

```

1:  $obs \leftarrow \emptyset$ 
2:  $\mathcal{L}_\beta \leftarrow$  child nodes of  $\eta$ 
3: if  $\mathcal{L}_\beta = \emptyset$  then
4:    $o \leftarrow$  observation of  $\eta$ 
5:    $t \leftarrow$  random timestamp from  $[t_{min}; t_{max}]$  of  $\eta$ 
6:   for  $i = 0$  to  $t$  do
7:     add  $o$  to  $obs$ 
8:   end for
9:   return  $obs$ 
10: end if
11:  $\eta^* \leftarrow$  random node from  $\mathcal{L}_\beta$ 
12: if  $\eta^*$  is a decomposition child of  $\eta$  then
13:    $o \leftarrow$  observations of  $\eta^*$ 
14:   add  $o$  to  $obs$ 
15: else
16:    $o \leftarrow getPrevObs(\eta)$ 
17:    $t \leftarrow$  random timestamp from  $[t_{min}; t_{max}]$  of  $\eta$ 
18:   for  $i = 0$  to  $t$  do
19:     add  $o$  to  $obs$ 
20:   end for
21:    $o^* \leftarrow getObservations(\eta^*)$ 
22:   add  $o^*$  to  $obs$ 
23: end if
24: return  $obs$ 

```

Final Remarks

In this chapter, we presented an overview of the temporal multiple-plan recognition architecture developed to deal with ambiguous and concurrent plans. We extended the SBR algorithm to disambiguate hypotheses based on information about time of day and duration of each plan execution acquired in previous episodes, apart from monitoring temporal errors in plan execution. We also described the dataset used to evaluate our recogniser approach. This dataset contains information about the activity duration carried out by the subject including start and end time, besides presenting a high-level

representation of all sequences of activities. The temporal plan library generator builds plan libraries based on given parameters and information about the minimum and the maximum number of timestamps required to perform the activity, which made it possible to include temporal aspects to plans in the library. The temporal input generator creates sequences of observations based on this temporal plan library. In Chapter 6, we evaluate SBR and PEC performances using both real and generated datasets.

6. EVALUATION

To evaluate Symbolic Behaviour Recognition (SBR) and Plan Execution Controller (PEC) performance, we used the real data collected by Tapia [Mun03], earlier presented in Section 5.3. In addition, we also performed experiments based on generated multiple temporal plan libraries using the Temporal Plan Library Generator (Section 5.4) alternating several parameters and getting temporal sequence of observations from these libraries through the Temporal Input Set Generator (Section 5.5). Several experiments were carried out in real and generated datasets in order to evaluate and compare the performances between SBR and PEC. Thus, for each dataset were performed 1000 complete recognition process¹ in plans chosen randomly. All experiments were conducted on a Dell Server with processor Intel® Xeon® E3-1220 v5 @ 3.00GHz × 4, graphic Gallium 0.4 on llvmpipe (LLVM 3.8, 256 bits), and 7.6 GiB of memory running Ubuntu 16.04.2 LTS 64-bit as base system.

6.1 Basic Evaluation Measures

In this section, we talk about evaluation metrics used to analyse results of the plan recognition performance between SBR and PEC. The PEC filter component can be evaluated as a binary classifier, because in a way it “labels” hypothesis as positive (*i.e.*, plans actually being performed by the user) and negative (*i.e.*, plans that are not being performed by the user). We also evaluate SBR hypotheses using these measures, since this recogniser either “labels” plans as positive and negative. That is, every hypothesis matching actions sequence performed by the user are classified as positive while all other plans (present in the plan library) are classified as negative (*i.e.*, SBR deems that those plans are not being currently executed by the user).

In binary decision problems, a classifier divides the dataset into two groups, producing output with two class values for given input data. The classifier then predicts all data instances as either **Positive (P)** or **Negative (N)**. This classification produces four types of outcome:

- **True Positives (TP):** are instances correctly classified as positive, in other words, in our context they refer to all cases where the algorithm recognised a plan that was really being performed by the user (*i.e.*, predicted positive, actual positive);
- **True Negatives (TN):** are instances correctly classified as negative, that is, all cases where the algorithm did not recognise a plan as a potential candidate and in

¹We consider a complete recognition process when all steps of a plan are performed successfully.

fact that plan was not being performed by the user (*i.e.*, predicted negative, actual negative);

- **False Positives (FP):** are negative instances incorrectly classified as positive, *i.e.*, all cases where the algorithm recognised a plan that was not being performed by the user (“*Type I error*” – predicted positive, actual negative);
- **False Negatives (FN):** are positive instances incorrectly classified as negative, *i.e.*, all cases where the algorithm did not recognise a plan which was being performed by the user (“*Type II error*” – predicted negative, actual positive).

These four outcomes are tabulated in a two by two (2×2) table called Confusion Matrix [Pow11] (Table 6.1). Thus, the decision made by the recogniser can be represented in this table used for calculating all basic evaluation measures of binary classification [BBF15]. We use these metrics for measuring the performance of plan recognition algorithms, comparing the SBR and PEC results through several recognition episodes.

		Actual	
		P	N
Predicted	P	TP	FP <i>Type I error</i>
	N	FN <i>Type II error</i>	TN

Table 6.1: Confusion matrix

The measures most utilised in this study are precision (PPV), recall (TPR), F_1 score, accuracy (ACC) [BBC⁺00], and false positive rate (FPR). Other basic evaluation measures are also calculated, such as Matthews correlation coefficient (MCC) [Mat75, LCY⁺15], error rate (ERR), true negative rate (TNR), false negative rate (FNR), negative predictive value (NPV), false discovery rate (FDR), and false omission rate (FOR). In the following, we briefly summarise these measures and their formulas.

- **Positive Predictive Value (PPV):** also known as *Precision*, denotes the proportion of instances predicted as positive that are actually positive instances (*i.e.*, the prediction was correct). It is determined by Equation 6.1, calculated as the number of true positives (TP) divided by the total number of positives predictions (TP + FP). The best PPV is 1, whereas the worst is 0.

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (6.1)$$

- **True Positive Rate (TPR):** also known as *Recall*, *Sensitivity*, and *Detection Probability*, is the proportion of true positive instances that are correctly predicted as positive. It is defined in Equation 6.2 using the number of true positives (TP) divided by the total number of positives (TP + FN). The best TPR is 1, whereas the worst is 0.

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (6.2)$$

- **True Negative Rate (TNR):** also known as *Specificity*, is the proportion of negative instances that are correctly predicted negative. It is calculated (Equation 6.3) using the number of true negatives (TN) divided by the total number of negatives (N). The best TNR is 1, whereas the worst is 0.

$$\text{TNR} = \frac{\text{TN}}{\text{N}} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (6.3)$$

- **Negative Predictive Value (NPV):** is the proportion of negatives instances that really are true negatives. It is defined in Equation 6.4 as the number of true negatives (TN) divided by the total number of negative predictions (TN + FN). The best NPV is 1, whereas the worst is 0.

$$\text{NPV} = \frac{\text{TN}}{\text{TN} + \text{FN}} \quad (6.4)$$

- **False Negative Rate (FNR):** also known as *Miss Rate*, is the proportion of positives (P) that are predicted as negative (N). Defined in Equation 6.5 is the complement of true positive rate (TPR). The best FNR is 0, whereas the worst is 1.

$$\text{FNR} = \frac{\text{FN}}{\text{P}} = \frac{\text{FN}}{\text{FN} + \text{TP}} = 1 - \text{TPR} \quad (6.5)$$

- **False Positive Rate (FPR):** also known as the *Fall-out*, is the complement of the true negative rate (TNR), *i.e.*, the proportion of negatives (N) that are incorrectly classified as positive (P). Defined in Equation 6.6, it is calculated as the number of false positives (FP) divided by the total number of negatives (N). The best FPR is 0, whereas the worst is 1.

$$\text{FPR} = \frac{\text{FP}}{\text{N}} = \frac{\text{FP}}{\text{FP} + \text{TN}} = 1 - \text{TNR} \quad (6.6)$$

- **False Discovery Rate (FDR):** is the complement of positive prediction value (PPV), *i.e.*, the proportion of false positives in the set of positive predictions. Defined in Equation 6.7 as the number of false positives (FP) divided by the total number of positives predictions (FP + TP). The best FDR is 0, whereas the worst is 1.

$$\text{FDR} = \frac{\text{FP}}{\text{FP} + \text{TP}} = 1 - \text{PPV} \quad (6.7)$$

- **False Omission Rate (FOR):** is the complement of negative predictive value (NPV), representing the proportions of negative instances that are false negatives. Defined in Equation 6.4 as the number of false negatives (FN) divided by the total number of negative predictions (FN + TN). The best FOR is 0, whereas the worst is 1.

$$\text{FOR} = \frac{\text{FN}}{\text{FN} + \text{TN}} = 1 - \text{NPV} \quad (6.8)$$

- **Accuracy (ACC):** is the fraction of predictions that are correct, represented in Equation 6.9 as the number of all correct predictions (TP + TN) divided by the total size of the dataset (P + N). The best ACC is 1, whereas the worst is 0.

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (6.9)$$

- **F₁ score:** is a harmonic mean of precision (PPV) and recall (TPR) balancing PPV and TPR equally and reducing them to Equation 6.10.

$$\text{F}_1 = 2 \times \frac{\text{PPV} \times \text{TPR}}{\text{PPV} + \text{TPR}} \quad (6.10)$$

- **Matthews Correlation Coefficient (MCC):** is essentially the correlation coefficient calculated, as presented in Equation 6.11, using all four values in the confusion matrix. It will return a value range in -1 to 1. The value of correlation coefficient -1 represents a classification that is always wrong, the value 0 represents an classification no better than random, and 1 when it is always correct.

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TP} + \text{FP})(\text{TN} + \text{FN})}} \quad (6.11)$$

- **Error Rate (ERR)** is the probability that the recogniser will incorrectly recognise an instance. Described in Equation 6.12 as the number of all incorrect predictions (FP + FN) divided by the total number of the dataset (TP + TN + FP + FN). Ideal performance is achieved (*i.e.*, the error rate is zero) when FN and FP are both zero.

$$\text{ERR} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (6.12)$$

These metrics can help us understand how a plan recognition algorithm performs. Moreover, we also use some plots to provide graphical representations because they are visually appealing and provide an overview of a recogniser performance across a wide range of specificities. The graphs used in this work are the Receiver Operating Characteristics (ROC space) [Faw06], Precision-Recall (PR space) [DG06] and Boxplot [MTL78].

- **Receiver Operating Characteristics (ROC space):** displays the variance between the number of instances correctly classified as positive (P) and the number of instances incorrectly classified as negative (N). The goal in ROC space is to reach as close to the top left corner as possible. In ROC space, one plots the true positive rate (TPR) on the y-axis and the false positive rate (FPR) on the x-axis. Thus, the y-axis measures the proportion of positive examples that are correctly classified and the x-axis measures the proportion of negative examples that are incorrectly classified as positive.
- **Precision-Recall (PR space):** shows precision values (PPV) for corresponding recall values (TPR). PR space plots the precision (PPV) on the y-axis and recall (TPR) on the x-axis. Thus, y-axis measures the proportion of instances classified as positive that are really positive whereas the x-axis measures the proportion of positive examples that are correctly classified. The goal in PR space is to reach as close to the top right corner as possible. These plots are recommended for datasets with large class imbalances.
- **Boxplot:** displays batches of data, using the follow values from a set of data: the median, the upper and lower hinges (quartiles), and the extremes [MTL78]. These plots are preferred over the mean and standard deviation for samples with extreme outliers and for population distributions that are asymmetric or irregularly

shaped [KA14]. Therefore, they are particularly useful for comparing distributions between several groups or sets of data, besides taking up less space. The skewness and the degree of dispersion in the data are indicated by the spacing between the different parts of the box.

6.2 Plan sequence and set of observations

In this work, we evaluate the recogniser every time a plan went to next step. Then, in the end of the recognition process, we compute the average result for metrics presented in Section 6.1 based on recogniser results obtained throughout this recognition process. Before introducing how we get observations to recognition episodes, it is essential to understand that plan steps can occur in complex combinations. In the real world, plans can happen sequentially, in parallel, or interleaved.

Sequential plans — plans that do not occur at the same time and do not overlap with one another. That is, plans that happen consecutively, one after another (*e.g.*, in Figure 6.1, plan A followed by plan B).

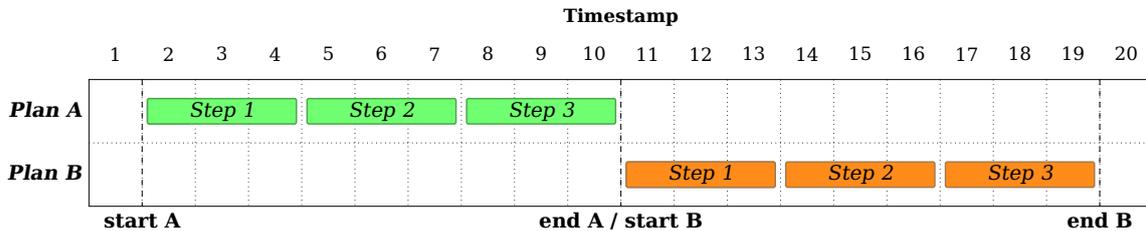


Figure 6.1: Example of sequential plans.

Parallel plans — plans that can be carried out simultaneously. In this work, we consider parallel activities as being in progress simultaneously but not necessarily involving full user’s attention/interaction at the same timestamp, such as watching TV and using the microwave. In Figure 6.2, plan B is parallel to plan A as from timestamp 6.

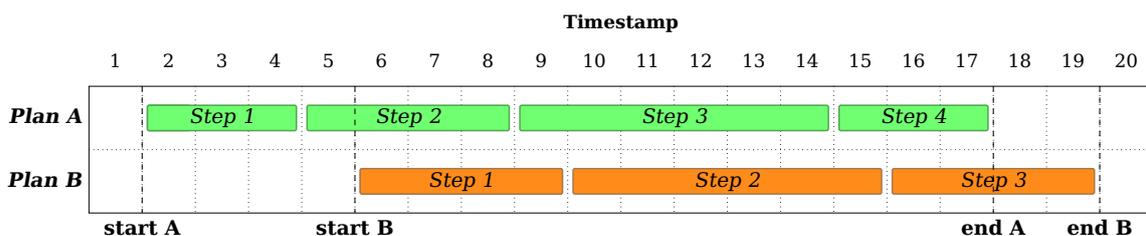


Figure 6.2: Example of parallel plans.

Interleaved plans — plans that the user switch between the execution of their plan steps within brief periods of time. Supposing the user is performing two plans, at each timestamp only one plan is identified as involving direct user interaction. The other plan at that timestamp is on hold, waiting for the user to return to its plan step currently in progress, that is, the plan has been started but not yet finished by the user. The main difference between sequential and interleaved plans is that user can return to finish one of the plan steps several times. In Figure 6.3, plans A and B are examples of interleaved plans.

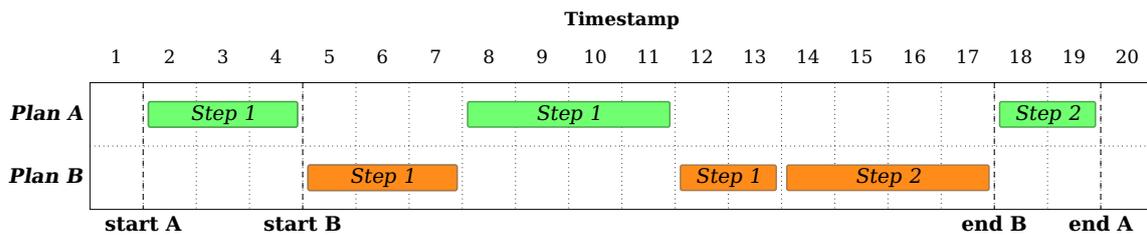


Figure 6.3: Example of interleaved plans.

Our temporal plan recognition aims to handle all these ways of performing multiples plans. Therefore, we carried out several simulations with one or more plans being executed at the same time. So we can evaluate the plan recogniser (SBR and PEC) each time the set of observations changes, that is, when a plan step starts or ends we check the plan(s) returned as hypotheses by the recogniser to evaluate its performance.

Figure 6.4 shows an example of three plans (A, B, and C) performed in parallel. Each blue number in parenthesis represents a recognition episode, *i.e.*, moments where the set of observations has changed. Thus, until the episode (1) only plan A was being performed, however, from now another plan has started and hence the recogniser must return also plan B as hypothesis. In episode (2), the set of observation has changed because B went on to its next activity (step 2). In this case, observations from the current step of A (step 2) remain the same but observations of step 1 of B are dropped and replaced by observations of step 2. In episode (3), another plan starts execution (plan C); from that moment and until episode (7) the user is performing three plans in parallel. We can see that there exists three episodes altering the observation set within that time interval. First, in episode (4) where plans A and C went to next step. Second, in the episode (5) when plans B and C went to next step and finally in the episode (6) where only plan A altered the set of observations. In all these episodes, ideally the plan recogniser should return these three plans as hypotheses and only these ones. The last episode (8) occurred in timestamp 19, representing the end of plans A and B, and consequently of the recognition process.

We call *recognition process* a particular simulation when a certain number of plans might be executed, possibly in parallel or interleaving. We call *recognition episode* each of the changes in observations within a recognition process.

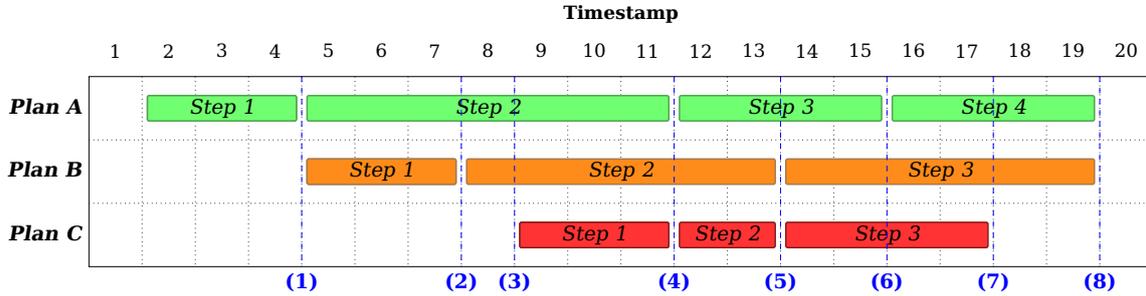


Figure 6.4: Example of a recognition process with various recognition episodes.

It is worth noting that we consider observations returned at each timestamp. Thus, for example, in Figure 6.4, the interval between timestamps 2 and 4 has three observation events returning the same set of observations formed by features (and their respective status) associated with the first step execution of plan A. At timestamp 5 two events happen to modify the set of observations, the plan A went to next step and plan B started. At the moments when the set of observation change, we evaluate the plan recognition algorithms (SBR and PEC) assuming as input the previous set of observations (in this case, the observations associated with the first step of A) and we compute the values for each of the metrics listed in Section 6.1 according to the hypotheses returned by the plan recogniser. This evaluation method is performed until the recognition process has finished. Thereupon, we compute the average for each metric acquired during the entire recognition process; that way, the recogniser performance is computed taking into account each step performed in the plan(s) execution. For example, suppose a user performing the example presented in Figure 6.4 and the precision (PPV) metric being calculated. In this case, for all eight recognition episodes² we calculate precision using the hypotheses returned by the recogniser. Then, in the end of recognition process, we compute the average precision using the eight precision values obtained during the recognition process. This way of evaluating performance allows us to check which recogniser is most efficient considering the entire recognition process.

6.3 Using a real-world dataset

In this work, we use the dataset files presented in Tapia [Mun03] where data is stored in CSV files in text format, presenting temporal information (e.g., start and

²Changes in the set of observations, i.e., episodes, are represented by blue numbers in parenthesis in Figure 6.4.

end time) about each activity performed by the subject. Besides, showing the sequence of activities related to the corresponding sequence of sensors activation. The dataset content was previously explained in Section 5.3, then in this section we focus on how a plan library is generated from this data.

As explained previously, we use the file `activities_data.csv` related with the subject to extract information required to generate a plan library. Table 6.2 shows data organisation in this file and an example of values representing its content. Based on this example we explain how dataset elements are used to generate a plan library.

Table 6.2: Example of data organisation and content in `activities_data.csv`

<code>activity_label</code>	<code>date</code>	<code>start_time</code>	<code>end_time</code>
<code>sensor1_id</code>	<code>sensor2_id</code>	...	
<code>sensor1_object</code>	<code>sensor2_object</code>	...	
<code>sensor1_activation_time</code>	<code>sensor2_activation_time</code>	...	
<code>sensor1_deactivation_time</code>	<code>sensor2_deactivation_time</code>	...	
Preparing breakfast	3/27/2003	6:50:8	6:57:56
80	143	55	...
Cabinet	Microwave	Cabinet	...
6:51:43	6:54:9	6:54:16	...
6:51:46	13:7:43	6:54:19	...

In Table 6.2 grey highlight line are elements used to create top-level plans, that is, `activity_label` is used to generate “id” and “label” for a plan/goal (e.g., top plan id and label will be “*Preparing_breakfast*”) while `start_time` and `end_time` are used to figure out the time of day the activity is usually performed. The other lines contain information to create plan steps, in this case, each column represents a plan step and consequently, the sum of columns is the number of steps needed to complete the plan path execution. Considering the first column representing the first plan step, `sensor1_id` is used together with `activity_label` to generate the plan step “id”. For instance, Table 6.2 first column, the plan step id will be *Preparing_breakfast_80*, next column, the second plan step id will be *Preparing_breakfast_80_143*, and so on. This method to generate “ids” is important to ensure a unique “id” for each plan step in a plan sequence.

The plan step label assumes the same `sensor_object` name (e.g., Cabinet, Microwave, etc.). It is worth noting that, dataset elements may have equal labels to different objects, according to columns one and three in Table 6.2, despite the same name (label) they represent distinct objects. Wherefore, we use `sensor_id` also as feature name to ensure a feature observation associated only with a specific object activation. The last two lines contain the plan step time information, so through the values of `sensor_activation_time` and `sensor_deactivation_time` we calculate plan step duration (e.g., Table 6.2 first column, the subject interacted with Cabinet for 3 seconds). The minimum and maximum plan step duration are acquired by travelling the table column, finding

plan steps with the same “id” and comparing the duration among each one to find the plan step minimum and maximum duration.

It is important to emphasise two points about generating a plan library from datasets. First, *sensor_deactivation_time* may not exceed the *end_time*. Whenever this occurs, we just consider *sensor_deactivation_time* assuming the same value of *end_time*. For example, in Table 6.2 the microwave only finish its execution (13:7:43) long after the plan *end_time* (6:57:56), in this case, *sensor_deactivation_time* will receive the value of *end_time*. Second, when current *sensor_deactivation_time* is greater than next step *sensor_activation_time* means that current object sensor keeps activated on next plan step. Then, current plan step feature will be required to enable recognition of next plan step, that way, current feature is associated with every next step presenting *sensor_activation_time* smaller than current step *sensor_deactivation_time*. For instance, according to Table 6.2, the microwave remains active for all next steps until the plan ended, so, the microwave status becomes a feature replicated in all next steps.

In this work, PEC components try to define the time of day an activity is usually performed, aiming to use this information to disambiguate hypotheses returned by SBR. Equation 5.1 is used to estimate this time, where the greater the number of times the plan was performed the greater the precision of this value. As previously explained, we store the number of times an activity has been successfully executed, identified as *nupd* (number of updates) in database. In these experiments, *nupd* represents how many times all paths for each plan was performed before the simulation start. This training stage is important to learn time information about each activity, such as minimum and maximum duration, besides the time of the day a plan usually happen.

Before analysing experiments results is important to emphasise we use two datasets storing sequences of activities performed by two different people (subject one and subject two) presented in Tapia [Mun03]. In this work, we called *Partial* the dataset where activities with less than six examples were eliminated. The number of examples per activity for each subject in partial dataset is shown in Table 5.4. The entire dataset was used in experiments, in this case, we called *Complete* dataset. The number of examples per activity for each subject in complete dataset is shown in Table 5.3.

It is worth noting that, to generate temporal plan libraries from a real dataset, first an average start time for each activity is calculated using all examples in dataset. After that, we check if some example of this activity has *start_time* outside the maximum range φ in relation to average start time. Thus, if some example is outside the range φ , then we tag activity hour as UNDEFINED. Otherwise, the activity hour is defined and receives average start time as its value in the temporal plan library. Another important point is that parallel plans must have similar activity hour, respecting the maximum time range φ among their hour. Only plans with their hour tagged as UNDEFINED can

be selected to perform in parallel with any other plan, in short, plans must be selected according to their hour.

In the experiments, we evaluate SBR and PEC performance in several datasets (real and generated) through different evaluation metrics. For each dataset, we performed 1000 complete recognition process in plan(s) chosen randomly. At the end of simulation, we compute mean for each evaluation metric and Mean Absolute Deviation (MAD) [Ami12] for some basic metrics. The MAD is a measure of dispersion that calculates how much the values in the dataset are likely to differ from their mean. The absolute value is used to avoid deviations with opposite signs cancelling each other out. The MAD is calculated using the following formula:

$$MAD = \frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}| \quad (6.13)$$

where n is the number of observed values, \bar{x} is the mean of the observed values and x_i are the individual values.

In this work, \mathcal{P} is the number of parallel plans being performed by the subject, for example, $\mathcal{P} = 1$ means the subject is performing a single plan, while $\mathcal{P} > 1$ means the subject is performing \mathcal{P} plans at the same time. The minimum path length δ is the minimum number of completed plan steps required for PEC to consider a candidate plan as a correct hypothesis. The next tables show mean (Mean) and mean absolute deviation (MAD) to the following evaluation metrics: precision (PPV), recall (TPR), F_1 score, accuracy (ACC) and false positive rate (FPR), besides the average time in nanoseconds (Time (ns)) to return the hypothesis set at each recognition episode.

6.4 Simulation results for subject one

The partial dataset of subject one contains the sequence of activities performed by a professional 30-year-old woman who spends free time at home, living alone in one-bedroom apartment in which were installed 77 state-change sensors collecting data for two weeks. In the partial dataset there are 13 top-level plans (see Table 5.4) presenting more than six execution paths with length varying from 1 to 88, *i.e.*, some plans are completed in a single plan step while others take up to 88 steps. The number of available features is equal the number of sensors present in apartment. In this case 77 features are available but they are not multi-valued, assuming only one value ('on') when activated, however, multiple features can appear in the same node in the plan library.

Tables 6.3 and 6.4 show simulation results for PEC and SBR performances, respectively, using subject one partial dataset with most plans tagged as undefined hour. Among all plans only *“going out to work”*, *“preparing lunch”*, *“preparing dinner”* and

“preparing breakfast” present a defined time of day to be executed. In Table 6.3, we observe that the greater the number of parallel plans \mathcal{P} the worse is the PEC performance. Just because many plans performing in parallel tend to increase ambiguity during recognition process making harder to the recogniser returns only plans that are actually being carried out. So, precision, recall and accuracy tend to present smaller values as the number of parallel plans increases in PEC algorithm. On the other hand, when minimum path length δ increases the PEC precision improve, showing that PEC algorithm presents other ways to disambiguate hypotheses set in addition to temporal constraints. When δ increases means the recogniser will wait for at least δ plan steps performed to consider this plan a candidate. So, plans at first steps probably will not be returned as hypothesis, which may decrease recall but increase precision significantly (Figure 6.6).

It’s worth noting that, as presented in Table 6.4, in the experiments SBR algorithm will always return all plan(s) being performed by the subject, so its recall always will be equal to 1, but precision tend to be smaller compared with PEC algorithm. In this case, only recall and execution time are better than PEC performance. Therefore, even in a real domain where most plans do not have a defined hour, containing plans with few steps and allowing parallel plans, our approach had better performance than SBR.

Table 6.3: PEC, subject one, partial dataset, most plans with undefined hour

\mathcal{P}	δ	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.667	0.191	0.776	0.151	0.685	0.126	0.929	0.034	0.058	0.040	1218761
	2	0.757	0.185	0.776	0.153	0.743	0.131	0.947	0.030	0.039	0.032	1255830
	3	0.886	0.144	0.719	0.215	0.816	0.121	0.965	0.024	0.014	0.019	1234891
2	1	0.577	0.141	0.753	0.130	0.635	0.109	0.858	0.055	0.127	0.061	1550325
	2	0.693	0.162	0.719	0.127	0.687	0.113	0.893	0.046	0.081	0.050	1541240
	3	0.844	0.152	0.684	0.137	0.741	0.118	0.929	0.034	0.033	0.033	1555201
3	1	0.554	0.118	0.714	0.125	0.611	0.098	0.796	0.063	0.189	0.072	1784764
	2	0.659	0.137	0.683	0.115	0.657	0.100	0.838	0.058	0.129	0.065	1744339
	3	0.823	0.137	0.623	0.107	0.696	0.096	0.885	0.041	0.054	0.043	1741836

Table 6.4: SBR, subject one, partial dataset, most plans with undefined hour

\mathcal{P}	δ	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.229	0.075	1.000	0.000	0.363	0.094	0.623	0.101	0.408	0.109	122224
	2	0.227	0.073	1.000	0.000	0.361	0.092	0.621	0.096	0.410	0.104	109289
	3	0.228	0.076	1.000	0.000	0.362	0.096	0.620	0.103	0.412	0.112	120262
2	1	0.261	0.056	1.000	0.000	0.409	0.069	0.544	0.091	0.529	0.108	241589
	2	0.262	0.054	1.000	0.000	0.411	0.067	0.549	0.085	0.523	0.101	229600
	3	0.262	0.055	1.000	0.000	0.411	0.067	0.552	0.089	0.520	0.105	212811
3	1	0.310	0.045	1.000	0.000	0.471	0.052	0.517	0.073	0.608	0.096	367108
	2	0.311	0.047	1.000	0.000	0.471	0.053	0.521	0.073	0.602	0.095	337625
	3	0.315	0.046	1.000	0.000	0.476	0.053	0.526	0.074	0.596	0.096	357386

Tables 6.5 and 6.6 show simulation results for PEC and SBR performances, using subject one partial dataset with all plans presenting a defined time of day to be executed. In Table 6.5, PEC presents similar behaviour to Table 6.3 in relation to variation

in both values \mathcal{P} and δ . However, presenting better performance in all metrics evaluated, showing that our approach is most adequate for domains where activities have a defined hour to perform. The SBR performance does not change by alternating δ value because this path length constraint is only used by PEC algorithm. In Table 6.6, SBR presents worse performance than PEC in most evaluation metrics (except recall). Thus, even with recall equal to 1, SBR harmonic mean of precision and recall (F_1 score) presents lower values when compared to PEC.

Table 6.5: PEC, subject one, partial dataset, all plans with defined hour

\mathcal{P}	δ	PPV		TPR		F_1 score		ACC		FPR		Time (<i>ns</i>)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.947	0.079	0.936	0.090	0.933	0.083	0.987	0.015	0.008	0.012	1086986
	2	0.962	0.060	0.912	0.124	0.939	0.078	0.988	0.015	0.005	0.009	1070132
	3	0.964	0.061	0.877	0.160	0.912	0.107	0.986	0.016	0.005	0.008	1173837
2	1	0.913	0.103	0.835	0.154	0.862	0.123	0.964	0.033	0.017	0.020	1278743
	2	0.929	0.094	0.812	0.171	0.860	0.129	0.965	0.033	0.013	0.017	1288032
	3	0.943	0.086	0.750	0.182	0.828	0.133	0.963	0.029	0.010	0.015	1336397
3	1	0.906	0.111	0.833	0.150	0.857	0.125	0.963	0.035	0.019	0.023	1276504
	2	0.918	0.105	0.803	0.175	0.854	0.131	0.962	0.036	0.015	0.020	1316860
	3	0.946	0.082	0.764	0.182	0.841	0.130	0.964	0.030	0.010	0.016	1310149

Table 6.6: SBR, subject one, partial dataset, all plans with defined hour

\mathcal{P}	δ	PPV		TPR		F_1 score		ACC		FPR		Time (<i>ns</i>)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.233	0.079	1.000	0.000	0.367	0.099	0.621	0.104	0.410	0.113	120488
	2	0.233	0.078	1.000	0.000	0.367	0.098	0.622	0.108	0.410	0.117	113739
	3	0.233	0.082	1.000	0.000	0.366	0.102	0.622	0.106	0.410	0.114	119907
2	1	0.247	0.064	1.000	0.000	0.388	0.079	0.582	0.100	0.471	0.120	181259
	2	0.247	0.064	1.000	0.000	0.389	0.080	0.579	0.099	0.475	0.119	205456
	3	0.244	0.067	1.000	0.000	0.383	0.083	0.573	0.102	0.481	0.123	200268
3	1	0.246	0.065	1.000	0.000	0.388	0.081	0.580	0.101	0.473	0.120	192238
	2	0.249	0.067	1.000	0.000	0.390	0.082	0.576	0.103	0.480	0.125	204341
	3	0.248	0.064	1.000	0.000	0.391	0.080	0.583	0.104	0.470	0.124	194751

The simulation results (Mean and MAD) for subject one using complete dataset with most plans tagged as undefined hour are shown in Tables A.1 and A.2. The results for this dataset with all plans presenting a defined hour are shown in Tables A.3 and A.4. The tables showing the average mean for each evaluation measures (PPV, TPR, F_1 score, ACC, ERR, MCC, FPR, TNR, FNR, NPV, FDR and FOR) computed in these experiments using subject one dataset are in Appendix A.

6.5 Simulation results for subject two

The complete dataset of subject two contains the sequence of activities performed by an 80-year-old woman who spent most of her time at home, living alone in

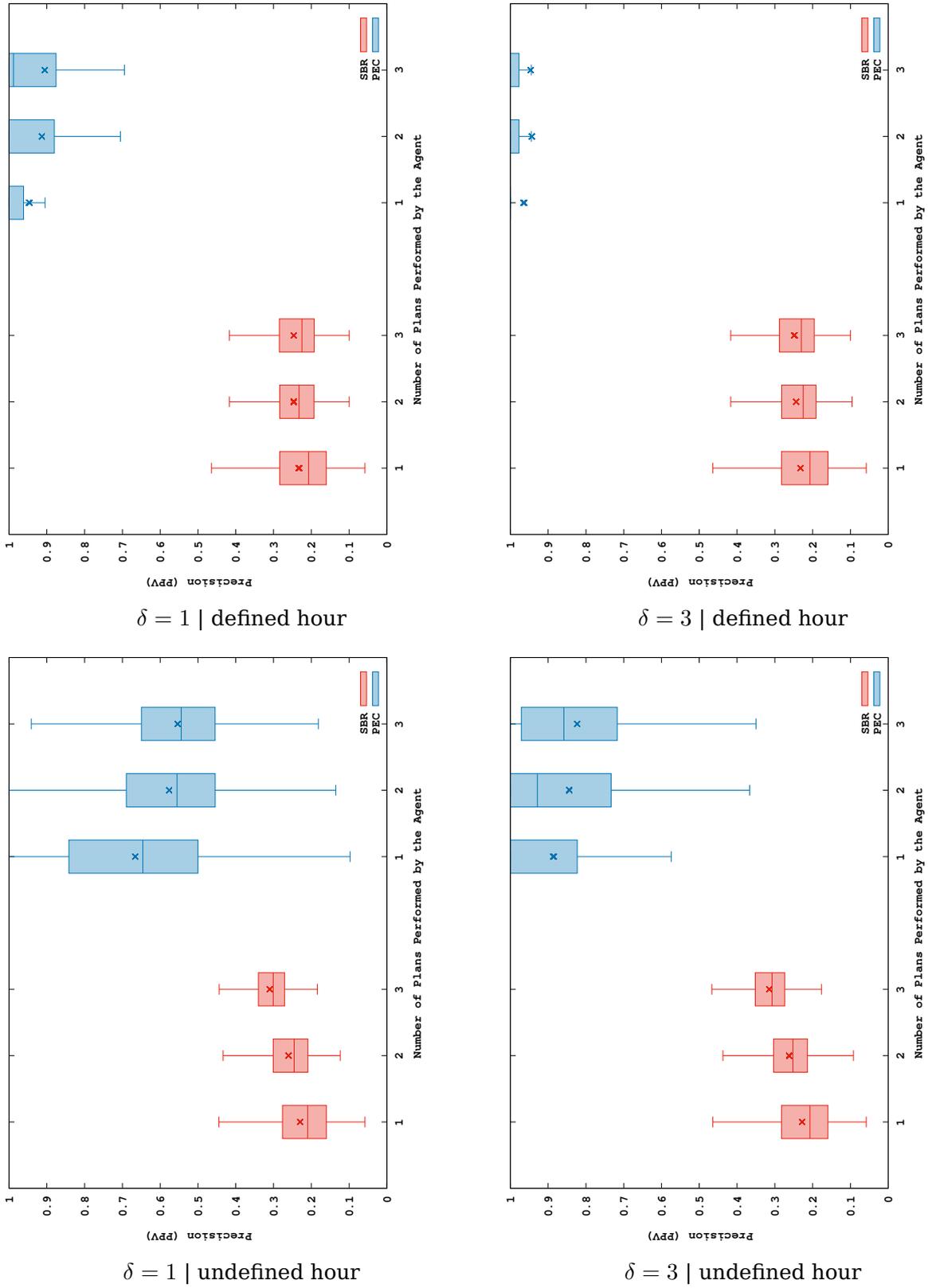


Figure 6.6: Boxplot graphs comparing SBR and PEC precision performance switching path length (δ) for subject one using partial dataset.

one-bedroom apartment in which were installed 84 state-change sensors collecting data for 14 days. In the complete dataset there are 208 execution paths distributed among 24 top-level plans (see Table 5.3) with length varying from 1 to 50.

Tables 6.7 and 6.8 show simulation results for PEC and SBR performances using subject two complete dataset with most plans tagged as undefined hour. Among all plans only “*preparing lunch*”, “*preparing dinner*” and “*preparing breakfast*” present a defined time of day to be executed and more than six execution paths. Other plans with defined hour have less than six execution paths, as “*Bathing*” and “*Going out for shopping*” have 3 execution paths each one, “*Other*” and “*Grooming*” with 2 paths, and “*Going out for entertainment*”, “*Lawnwork*”, “*Preparing a beverage*”, “*Putting away laundry*”, and “*Putting away groceries*” have only a single execution path. Although half of plans have defined hour, the number of execution paths with defined hour still is small.

In Table 6.7, we note a sharp decrease in recall values when δ increase compared to subject one with undefined hour (Table 6.3). The recall decrease may be because this dataset has 69 plans with less than three steps, that is, the dataset has many short plans. Thus, in this case, plans with less than three steps may be ignored as a valid hypothesis by PEC algorithm when $\delta > 1$ causing the recall decrease. Even in a dataset presenting many plans with an undefined hour and short paths, PEC results remain better than SBR comparing Tables 6.7 and 6.8.

Table 6.7: PEC, subject two, complete dataset, most plans with undefined hour

\mathcal{P}	δ	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.661	0.257	0.779	0.182	0.687	0.187	0.960	0.027	0.032	0.026	1600281
	2	0.785	0.219	0.689	0.229	0.731	0.150	0.970	0.019	0.018	0.020	1585851
	3	0.871	0.154	0.597	0.336	0.810	0.132	0.977	0.015	0.007	0.009	1554901
2	1	0.505	0.175	0.743	0.142	0.578	0.143	0.901	0.047	0.088	0.049	1975257
	2	0.686	0.204	0.697	0.151	0.663	0.133	0.937	0.032	0.045	0.035	1935193
	3	0.792	0.173	0.595	0.209	0.666	0.152	0.951	0.021	0.022	0.020	2039119
3	1	0.462	0.129	0.739	0.118	0.552	0.108	0.852	0.055	0.140	0.062	2317015
	2	0.594	0.151	0.696	0.119	0.619	0.103	0.894	0.040	0.087	0.045	2315691
	3	0.750	0.161	0.548	0.153	0.608	0.121	0.920	0.027	0.039	0.029	2293678

Table 6.8: SBR, subject two, complete dataset, most plans with undefined hour

\mathcal{P}	δ	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.248	0.132	1.000	0.000	0.370	0.153	0.755	0.086	0.256	0.090	203249
	2	0.239	0.126	1.000	0.000	0.361	0.148	0.753	0.088	0.258	0.092	222329
	3	0.237	0.124	1.000	0.000	0.359	0.147	0.749	0.087	0.262	0.091	726215
2	1	0.227	0.070	1.000	0.000	0.360	0.087	0.679	0.073	0.346	0.079	479012
	2	0.237	0.080	1.000	0.000	0.371	0.097	0.685	0.077	0.340	0.084	480977
	3	0.227	0.071	1.000	0.000	0.360	0.088	0.675	0.073	0.350	0.080	519392
3	1	0.245	0.051	1.000	0.000	0.389	0.064	0.637	0.060	0.407	0.068	782808
	2	0.246	0.052	1.000	0.000	0.390	0.064	0.635	0.060	0.409	0.068	845159
	3	0.246	0.050	1.000	0.000	0.390	0.062	0.639	0.060	0.405	0.068	829582

Tables 6.9 and 6.10 show simulation results for PEC and SBR respectively, using subject two complete dataset with all plans presenting a defined time of day to be executed. In Table 6.9, PEC presents better results compared to Table 6.7 confirming that our approach is adequate to domains where plans perform in defined hour (Figure 6.8). However, when compared to Table 6.5 the results are worse due to the number of short plans, that is, dataset of subject one has only 37 paths with less than 3 steps in 280 plan paths, while dataset of subject two has 69 paths with less than 3 steps in 208 plan paths. Thus, dataset of subject two presents more short plans than dataset of subject one, consequently, analysing the tables content, we confirm that our approach works better in domains with longer plan paths.

Table 6.9: PEC, subject two, complete dataset, all plans with defined hour

\mathcal{P}	δ	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.942	0.095	0.901	0.128	0.912	0.108	0.992	0.010	0.004	0.006	1414605
	2	0.980	0.038	0.822	0.225	0.924	0.096	0.991	0.011	0.001	0.003	1429505
	3	0.990	0.020	0.733	0.292	0.899	0.115	0.988	0.012	0.001	0.001	1470727
2	1	0.813	0.195	0.801	0.177	0.793	0.162	0.973	0.023	0.017	0.018	1646426
	2	0.873	0.148	0.732	0.200	0.783	0.144	0.974	0.017	0.011	0.014	1769721
	3	0.949	0.080	0.636	0.272	0.807	0.138	0.976	0.017	0.004	0.006	1666964
3	1	0.733	0.215	0.745	0.184	0.719	0.166	0.956	0.029	0.030	0.026	1820602
	2	0.835	0.161	0.701	0.200	0.755	0.136	0.963	0.025	0.019	0.020	1919968
	3	0.877	0.132	0.630	0.255	0.764	0.151	0.965	0.024	0.012	0.014	1853750

Table 6.10: SBR, subject two, complete dataset, all plans with defined hour

\mathcal{P}	δ	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.244	0.129	1.000	0.000	0.366	0.149	0.755	0.087	0.256	0.091	412081
	2	0.245	0.130	1.000	0.000	0.367	0.151	0.756	0.085	0.255	0.089	185169
	3	0.239	0.126	1.000	0.000	0.361	0.147	0.752	0.085	0.259	0.088	217185
2	1	0.231	0.099	1.000	0.000	0.359	0.120	0.717	0.086	0.300	0.092	354345
	2	0.222	0.090	1.000	0.000	0.349	0.110	0.705	0.083	0.314	0.090	393015
	3	0.224	0.088	1.000	0.000	0.352	0.109	0.718	0.086	0.300	0.093	365251
3	1	0.225	0.080	1.000	0.000	0.355	0.101	0.695	0.089	0.329	0.102	470355
	2	0.233	0.080	1.000	0.000	0.366	0.100	0.705	0.090	0.318	0.103	518653
	3	0.224	0.078	1.000	0.000	0.355	0.099	0.696	0.087	0.328	0.099	511486

The simulation results (Mean and MAD) for subject two using partial dataset with most plans tagged as undefined hour are shown in Tables B.1 and B.2. The results for this dataset with all plans presenting a defined hour are shown in Tables B.3 and B.4. The tables showing the average mean for each evaluation measures (PPV, TPR, F₁ score, ACC, ERR, MCC, FPR, TNR, FNR, NPV, FDR and FOR) computed in these experiments using subject two dataset are in Appendix B.

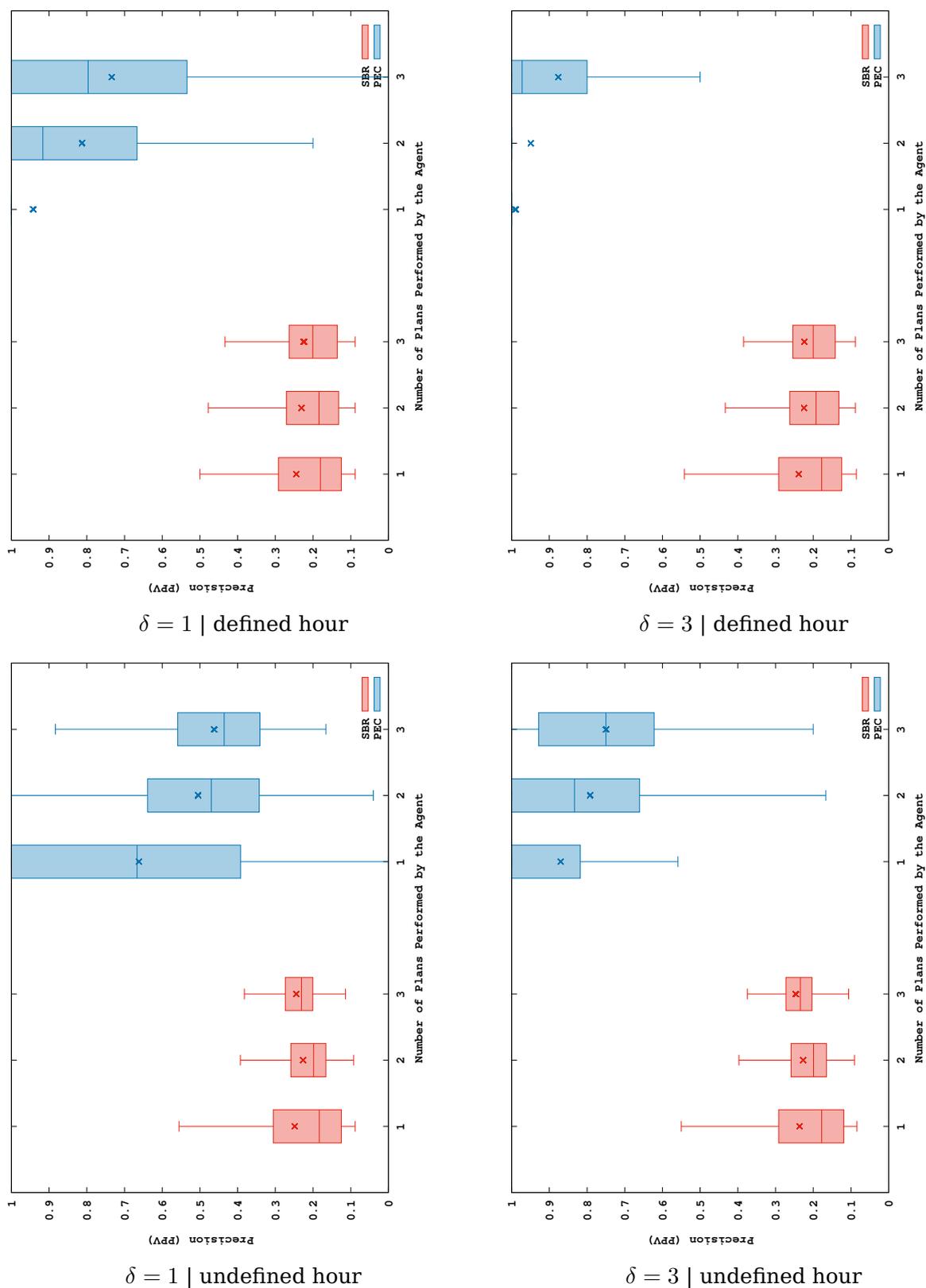


Figure 6.8: Boxplot graphs comparing SBR and PEC precision performance switching path length (δ) for subject two using complete dataset.

6.6 Simulation results using generated plan libraries

The simulations based on synthetically generated plan library aim to compare SBR and PEC performance in controlled datasets created from determined parameters and presenting specific data structure, so, experiments allow to evaluate a recogniser performance in different domains. Therefore, an extensive set of experiments varying different parameters was carried out in order to evaluate SBR and PEC results. We use the *Temporal Plan Library Generator* (Section 5.4) to build a temporal plan library based on the given parameters and the *Temporal Input Set Generator* (Section 5.5) to generate temporal sequences of observations based on the plan library.

In all experiments, we evaluated SBR and PEC results computing the evaluation measures (Section 6.1), obtained from 1000 simulation performed using random temporal observations based on the given temporal plan library. Some parameters in temporal plan library generator are fixed in the experiments, such as: sequential edges probability ($\Phi = 1$), duplication probability ($\Psi = 0$), and minimum ($t = 1$) and maximum ($T = 100$) number of observations. Thus, in this case, we only have sequential paths without any mechanism to generate ambiguity in the plan library and plan steps can not take more than 100 timestamps to complete. Another important information is that each plan/goal has a specific hour from 0 to 23, in which usually is performed, or the plan is tagged as UNDEFINED with a probability of 4%. So, in simulations most plans have a specific time of day to be performed. It's worth noting that plan start hour is not the same in all simulations since this hour may be increased/decreased by range value φ used to simulate different start times (in this work φ is equal 900 seconds).

In this section we analyse SBR and PEC performance in different domains generated by temporal plan library generator varying several parameters. Tables 6.11 and 6.12 show part of simulation results by altering the following parameters: number of top-level plans g (25 – 100), depth of the plan trees d (5 – 11), and available features F (10 – 50). Other parameters remain fixed as the number of parallel plans ($\mathcal{P} = 1$), minimum and maximum number of branches ($b = 1$ and $B = 2$), number of features per node ($f = 1$), and number of feature status ($c = 2$).

In Table 6.11, we note that PEC presents the worst result when there are many top-level plans ($g = 100$), short plans ($d = 5$), and few features available in domain ($F = 10$). In this case, even precision presenting a satisfactory result, the recall is relatively lower compared to other values. We also observe that a high number of top plans with few available features increase the recognition runtime. In Table 6.12, we notice that SBR also presents the worst precision when there are many top-level plans ($g = 100$) and few features ($F = 10$). However, path length (d) has no significant influence on recogniser performance, these results also show a strong association between number

of top-plans and number of features in SBR performance, that is, a lower number of top-plans and a greater number of features improve SBR performance.

Table 6.11: PEC results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$d$	$\$F$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	5	10	0.988	0.021	0.885	0.124	0.929	0.071	0.995	0.005	0.001	0.002	605000
		30	0.994	0.012	0.995	0.010	0.994	0.012	0.999	0.001	0.001	0.001	272500
		50	0.989	0.021	0.993	0.015	0.990	0.018	0.999	0.002	0.001	0.002	285000
	8	10	0.968	0.044	0.970	0.047	0.968	0.038	0.996	0.004	0.002	0.003	474286
		30	0.979	0.032	0.984	0.028	0.981	0.027	0.998	0.003	0.002	0.003	311429
		50	0.997	0.005	0.994	0.011	0.995	0.008	1.000	0.001	0.000	0.000	234286
	11	10	0.992	0.015	0.978	0.034	0.984	0.023	0.998	0.002	0.001	0.001	482000
		30	0.977	0.028	0.997	0.006	0.987	0.016	0.998	0.002	0.002	0.002	222000
		50	0.995	0.009	0.999	0.002	0.997	0.006	1.000	0.001	0.000	0.001	217000
100	5	10	0.953	0.077	0.698	0.186	0.782	0.149	0.996	0.002	0.001	0.001	2022500
		30	0.995	0.010	0.948	0.085	0.966	0.052	0.999	0.001	0.000	0.000	712500
		50	0.989	0.021	0.965	0.060	0.974	0.041	0.999	0.001	0.000	0.000	527500
	8	10	0.971	0.042	0.867	0.069	0.913	0.047	0.998	0.001	0.000	0.001	1777143
		30	0.957	0.054	0.960	0.058	0.957	0.046	0.999	0.001	0.001	0.001	737143
		50	0.987	0.022	0.984	0.028	0.985	0.023	1.000	0.001	0.000	0.000	515714
	11	10	0.982	0.027	0.930	0.056	0.954	0.030	0.999	0.001	0.000	0.000	1834000
		30	0.996	0.007	0.975	0.038	0.985	0.022	1.000	0.000	0.000	0.000	733000
		50	0.992	0.014	0.983	0.028	0.987	0.019	1.000	0.000	0.000	0.000	471000

Table 6.12: SBR results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$d$	$\$F$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	5	10	0.456	0.093	1.000	0.000	0.618	0.086	0.928	0.018	0.075	0.019	40000
		30	0.774	0.115	1.000	0.000	0.866	0.073	0.977	0.015	0.024	0.015	17500
		50	0.857	0.078	1.000	0.000	0.920	0.047	0.988	0.007	0.013	0.007	47500
	8	10	0.498	0.077	1.000	0.000	0.658	0.069	0.934	0.018	0.068	0.019	38571
		30	0.730	0.076	1.000	0.000	0.841	0.052	0.974	0.007	0.027	0.008	31429
		50	0.832	0.077	1.000	0.000	0.905	0.047	0.986	0.007	0.015	0.007	50000
	11	10	0.497	0.078	1.000	0.000	0.658	0.070	0.935	0.013	0.068	0.014	77000
		30	0.762	0.067	1.000	0.000	0.862	0.043	0.978	0.007	0.023	0.007	43000
		50	0.851	0.059	1.000	0.000	0.918	0.035	0.987	0.006	0.013	0.006	45000
100	5	10	0.113	0.018	1.000	0.000	0.203	0.028	0.912	0.012	0.088	0.012	67500
		30	0.346	0.071	1.000	0.000	0.507	0.076	0.974	0.005	0.026	0.005	50000
		50	0.523	0.108	1.000	0.000	0.677	0.093	0.984	0.005	0.016	0.005	60000
	8	10	0.120	0.010	1.000	0.000	0.214	0.015	0.921	0.007	0.080	0.007	185714
		30	0.343	0.060	1.000	0.000	0.506	0.067	0.974	0.005	0.027	0.005	82857
		50	0.502	0.082	1.000	0.000	0.663	0.072	0.984	0.004	0.016	0.004	62857
	11	10	0.127	0.013	1.000	0.000	0.225	0.020	0.921	0.007	0.079	0.007	329000
		30	0.358	0.054	1.000	0.000	0.524	0.059	0.974	0.004	0.026	0.004	156000
		50	0.565	0.081	1.000	0.000	0.717	0.067	0.986	0.003	0.014	0.003	118000

The tables showing complete results (Mean and MAD) for PEC and SBR applied to recognise one, two or three parallel plan(s) (\mathcal{P}) in generated simulations alternating the number of top plans ($\$g$), depth ($\d), and available features ($\$F$) are shown in Tables C.1, C.2, C.3, C.4, C.5, and C.6.

Tables 6.13 and 6.14 show part of simulation results by varying the following parameters: number of top-level plans $\$g$ (25 – 100), number of features per node $\$f$ (1 – 3), and number of feature status $\$c$ (1 – 3). Other parameters remain fixed as the number of parallel plans ($\mathcal{P} = 3$), minimum and maximum number of branches ($\$b = 1$ and $\$B = 2$), depth of the plan trees ($\$d = 11$), and available features ($\$F = 10$).

In Table 6.13, the worst PEC results are when there are several top-level plans ($\$g = 100$), few features per node ($\$f = 1$), and features are not multi-valued, *i.e.*, features can assume only one status ($\$c = 1$). The precision value still is high considering that 3 plans are performing in parallel, but recall is very lower showing that on average less than half of correct plans are returned by the recogniser. These parameters also increase considerably the recognition runtime. In Table 6.14 SBR presents precision values extremely low in domains with many top-level plans ($\$g = 100$), few features per node ($\$f = 1$), and principally when features are not multi-valued ($\$c = 1$). The results reveal the great dependence of the SBR with the number of features, its status variety and quantity per node, showing that in domains with many plans a large number of multi-valued features is essential to SBR present a better precision, otherwise, will be very hard for this recogniser return only the correct plan(s).

Table 6.13: PEC results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$f$	$\$c$	PPV		TPR		F_1 score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	1	1	0.958	0.054	0.869	0.101	0.908	0.079	0.986	0.013	0.003	0.004	1810098
		2	0.996	0.008	0.930	0.057	0.960	0.033	0.994	0.005	0.000	0.001	840305
		3	0.934	0.059	0.903	0.054	0.916	0.045	0.984	0.009	0.008	0.007	931409
	2	1	0.988	0.018	0.960	0.038	0.974	0.025	0.995	0.005	0.002	0.002	1026496
		2	0.999	0.001	0.970	0.031	0.984	0.016	0.997	0.003	0.000	0.000	415745
		3	1.000	0.000	0.986	0.017	0.993	0.009	0.999	0.001	0.000	0.000	319222
	3	1	0.943	0.048	0.906	0.067	0.922	0.054	0.982	0.013	0.009	0.008	1722839
		2	0.979	0.022	0.980	0.018	0.979	0.016	0.997	0.003	0.002	0.002	367649
		3	0.994	0.010	0.981	0.016	0.988	0.010	0.998	0.002	0.000	0.001	278117
100	1	1	0.732	0.133	0.436	0.116	0.539	0.121	0.980	0.004	0.004	0.002	10363800
		2	0.897	0.062	0.754	0.081	0.815	0.068	0.992	0.003	0.002	0.001	4115618
		3	0.909	0.055	0.817	0.053	0.858	0.044	0.994	0.002	0.002	0.001	2876787
	2	1	0.832	0.072	0.729	0.094	0.772	0.073	0.988	0.004	0.005	0.002	7439297
		2	0.938	0.040	0.892	0.039	0.913	0.030	0.996	0.001	0.001	0.001	1652656
		3	0.959	0.023	0.929	0.032	0.943	0.019	0.998	0.001	0.001	0.001	915113
	3	1	0.833	0.068	0.749	0.097	0.784	0.078	0.987	0.004	0.006	0.003	9143203
		2	0.947	0.034	0.935	0.026	0.940	0.022	0.998	0.001	0.001	0.001	1000641
		3	0.983	0.015	0.981	0.016	0.982	0.011	0.999	0.000	0.000	0.000	477340

The complete results (Mean and MAD) for PEC and SBR applied to recognise one, two or three parallel plan(s) (\mathcal{P}) in simulations varying the number of top plans ($\$g$), features per node ($\f), and feature status ($\$c$) are shown in Tables C.7, C.8, C.9, C.10, C.11, and C.12.

Tables 6.15 and 6.16 show part of simulation results of PEC and SBR respectively, by varying the following parameters: number of available features $\$F$ (10 – 50),

Table 6.14: SBR results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$f$	$\$c$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	1	1	0.203	0.016	1.000	0.000	0.338	0.022	0.700	0.096	0.328	0.112	226287
		2	0.455	0.080	1.000	0.000	0.619	0.074	0.897	0.038	0.111	0.042	106635
		3	0.484	0.051	1.000	0.000	0.650	0.046	0.891	0.018	0.119	0.020	114783
	2	1	0.509	0.130	1.000	0.000	0.662	0.112	0.878	0.077	0.134	0.086	128724
		2	0.738	0.150	1.000	0.000	0.839	0.098	0.963	0.025	0.039	0.027	68210
		3	0.852	0.090	1.000	0.000	0.916	0.053	0.985	0.010	0.016	0.011	40736
	3	1	0.390	0.084	1.000	0.000	0.555	0.085	0.736	0.109	0.298	0.126	272426
		2	0.758	0.073	1.000	0.000	0.860	0.047	0.974	0.010	0.028	0.010	80674
		3	0.829	0.047	1.000	0.000	0.905	0.028	0.984	0.005	0.017	0.005	64638
100	1	1	0.061	0.004	1.000	0.000	0.116	0.007	0.540	0.027	0.474	0.028	2500439
		2	0.106	0.013	1.000	0.000	0.191	0.021	0.798	0.013	0.206	0.013	612724
		3	0.137	0.016	1.000	0.000	0.240	0.025	0.866	0.011	0.137	0.012	453216
	2	1	0.099	0.013	1.000	0.000	0.179	0.021	0.657	0.032	0.354	0.033	1400142
		2	0.226	0.037	1.000	0.000	0.367	0.049	0.926	0.009	0.075	0.009	279192
		3	0.372	0.046	1.000	0.000	0.540	0.048	0.967	0.004	0.034	0.004	140806
	3	1	0.104	0.016	1.000	0.000	0.188	0.026	0.594	0.046	0.418	0.047	1965743
		2	0.360	0.038	1.000	0.000	0.527	0.042	0.962	0.004	0.039	0.005	192750
		3	0.613	0.055	1.000	0.000	0.758	0.043	0.987	0.003	0.013	0.003	129861

number of features per node $\$f$ (1 – 3), and number of feature status $\$c$ (1 – 3). Other parameters remain fixed as the number of parallel plans ($\mathcal{P} = 3$), number of top-level plans ($\$g = 100$), minimum and maximum number of branches ($\$b = 1$ and $\$B = 2$), and depth of the plan trees ($\$d = 11$).

Table 6.15 shows that increasing anyone of these three parameters, the number of available features $\$F$, their number of valid status $\$c$ or the number of features per node $\$f$ improves PEC performance in all metrics evaluated, especially recall and recognition time. Analysing Table 6.16 content is notable the great influence that parameters related to the features exert on SBR, and it is worth noting the improvement in precision values that are directly related to decreasing of false positive rate (FPR).

The complete results (Mean and MAD) for PEC and SBR applied to recognise parallel plan(s) (\mathcal{P} equal to 1, 2 or 3) in several simulations varying the number of features ($\$F$), features per node ($\f), and feature status ($\$c$) are shown in Tables C.13, C.14, C.15, C.16, C.17, and C.18. The tables showing the average mean for each evaluation measures (PPV, TPR, F₁ score, ACC, ERR, MCC, FPR, TNR, FNR, NPV, FDR and FOR) computed in these experiments using synthetically generated plan libraries are in Appendix C.

Final Remarks

In this chapter, we evaluated the Symbolic Behaviour Recognition (SBR) and the Plan Execution Controller (PEC) performance, using real-world and synthetically gener-

Table 6.15: PEC results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

\\$F	\\$f	\\$c	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
10	1	1	0.732	0.133	0.436	0.116	0.539	0.121	0.980	0.004	0.004	0.002	10363800
		2	0.897	0.062	0.754	0.081	0.815	0.068	0.992	0.003	0.002	0.001	4115618
		3	0.909	0.055	0.817	0.053	0.858	0.044	0.994	0.002	0.002	0.001	2876787
	2	1	0.832	0.072	0.729	0.094	0.772	0.073	0.988	0.004	0.005	0.002	7439297
		2	0.938	0.040	0.892	0.039	0.913	0.030	0.996	0.001	0.001	0.001	1652656
		3	0.959	0.023	0.929	0.032	0.943	0.019	0.998	0.001	0.001	0.001	915113
	3	1	0.833	0.068	0.749	0.097	0.784	0.078	0.987	0.004	0.006	0.003	9143203
		2	0.947	0.034	0.935	0.026	0.940	0.022	0.998	0.001	0.001	0.001	1000641
		3	0.983	0.015	0.981	0.016	0.982	0.011	0.999	0.000	0.000	0.000	477340
50	1	1	0.938	0.035	0.862	0.060	0.894	0.044	0.994	0.002	0.002	0.001	2077771
		2	0.985	0.019	0.899	0.062	0.935	0.042	0.997	0.002	0.001	0.001	1252194
		3	0.953	0.029	0.951	0.031	0.951	0.025	0.997	0.001	0.002	0.001	1010258
	2	1	0.984	0.013	0.970	0.026	0.975	0.019	0.999	0.001	0.001	0.000	896138
		2	0.991	0.011	0.978	0.029	0.982	0.019	0.999	0.001	0.000	0.000	530598
		3	0.998	0.004	0.997	0.005	0.997	0.004	1.000	0.000	0.000	0.000	444852
	3	1	0.987	0.012	0.977	0.028	0.980	0.019	0.999	0.001	0.001	0.000	730433
		2	0.994	0.009	0.994	0.010	0.994	0.007	1.000	0.000	0.000	0.000	459701
		3	0.997	0.006	0.998	0.004	0.997	0.004	1.000	0.000	0.000	0.000	383581

Table 6.16: SBR results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

\\$F	\\$f	\\$c	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
10	1	1	0.061	0.004	1.000	0.000	0.116	0.007	0.540	0.027	0.474	0.028	2500439
		2	0.106	0.013	1.000	0.000	0.191	0.021	0.798	0.013	0.206	0.013	612724
		3	0.137	0.016	1.000	0.000	0.240	0.025	0.866	0.011	0.137	0.012	453216
	2	1	0.099	0.013	1.000	0.000	0.179	0.021	0.657	0.032	0.354	0.033	1400142
		2	0.226	0.037	1.000	0.000	0.367	0.049	0.926	0.009	0.075	0.009	279192
		3	0.372	0.046	1.000	0.000	0.540	0.048	0.967	0.004	0.034	0.004	140806
	3	1	0.104	0.016	1.000	0.000	0.188	0.026	0.594	0.046	0.418	0.047	1965743
		2	0.360	0.038	1.000	0.000	0.527	0.042	0.962	0.004	0.039	0.005	192750
		3	0.613	0.055	1.000	0.000	0.758	0.043	0.987	0.003	0.013	0.003	129861
50	1	1	0.281	0.022	1.000	0.000	0.438	0.027	0.918	0.007	0.084	0.008	274620
		2	0.455	0.042	1.000	0.000	0.624	0.040	0.963	0.005	0.038	0.005	161793
		3	0.531	0.053	1.000	0.000	0.691	0.046	0.974	0.004	0.027	0.004	139791
	2	1	0.724	0.039	1.000	0.000	0.839	0.026	0.985	0.003	0.015	0.003	178608
		2	0.889	0.033	1.000	0.000	0.941	0.019	0.996	0.001	0.004	0.001	138155
		3	0.946	0.029	1.000	0.000	0.972	0.015	0.998	0.001	0.002	0.001	129759
	3	1	0.859	0.032	1.000	0.000	0.923	0.018	0.994	0.002	0.007	0.002	264337
		2	0.897	0.038	1.000	0.000	0.945	0.022	0.997	0.001	0.003	0.001	135416
		3	0.928	0.039	1.000	0.000	0.962	0.021	0.998	0.001	0.002	0.001	104510

ated datasets. We also discussed the evaluation metrics used to analyse the results of the plan recognition performance between SBR and PEC. Simulations in real and generated domains show a significant improvement of our approach in most evaluation metrics compared to SBR, especially in the precision measure, without any major decrease in recall. In Chapter 7, we survey related work about temporal and concurrent plan recognition.

7. RELATED WORK

There are only a few approaches that consider the problem of recognising activities and plans that are not necessarily sequential and uninterrupted. Furthermore, most of these approaches do not consider activities performing concurrently nor they use duration of each action aiming to improve the recognition process or to detect possible failures in the path of plan execution. Helaoui et al. [HNS11] make this less restrictive and more challenging assumption for activity recognition, proposing the use of *Markov Logic Networks* (MLN) [RD06] as a declarative framework for recognising interleaved and concurrent activities in *Ambient Assisted Living* (AAL) environments, incorporating both input from pervasive light-weight sensor technology and common-sense background knowledge.

Many scientific and engineering applications have studied and used *Hidden Markov Models* (HMMs) and their diverse variants successfully applied to a number of problems in several areas. However, the use of HMMs in analysing behavioural data for activity recognition is relatively recent and the duration of states is not modelled explicitly in conventional HMMs. Instead, transitions from a node to itself are used to allow a system to remain in its current state and an exponential state duration distribution is implicitly created when these self-transitions occur, unfortunately, this has presented some problems in recognising real-world activities, because HMMs are simple and efficient models for learning sequential data, but its performance tends to degrade when tasks present long-term temporal dependency or the range of tasks becomes more complex, these are inherent characteristics of real-world applications and difficult to deal with under the Markov assumption [DBPV05].

There are some methods for matching data to models in use for activity and plan recognition, HMMs are the basis for a series of these methods, particularly HMM variations [DPBV09] that take duration explicitly into account have become more popular, due to the strong temporal component present in many cases of recognition tasks. *Hidden Semi-Markov Models* (HSMMs) [HN03] and *Switching Hidden Semi-Markov Models* (S-HSMM) [DBPV05] are examples of such methods. These approaches allow explicit modelling of the length of time the observed agent is expected to be in a given state, in other words, the duration of a particularly recognised state. However, these models encounter difficulty to model distinctions for changes in transition probabilities based on the duration an agent has spent in a given state. Thus, they are insufficient for the complex set of recognition problems found in realistic applications.

Marhasev et al. [MHKF09] deal with low-likelihood hypotheses based on both duration and state in the context of a clinical diagnosis task. Investigating the use of Markov models that explicitly model the dependency of transition probabilities on state duration and presenting the *Non-stationary Hidden Semi-Markov Models* (NSHSMMs)

that is an augmentation of the standard semi-Markov models, which permit more precise recognition of duration-based behavioural patterns. Thus, as HSMMs expand HMMs to more accurately modelling realistically observed actions by including state duration, so do NSHSMMs expand HSMMs to represent the transition probabilities' dependence on state duration.

Duong et al. [DBPV05] introduce S-HSMMs to represent user activities that have a known duration. S-HSMMs are basically two-layered extensions of the HSMM, for abnormality detection and high-level activity recognition. Duong et al. included a formal definition of S-HSMMs, parameter estimation methods, and experimental validation for the innovative model, demonstrating the S-HSMMs superiority over the flat HSMM when applied to the task of abnormality detection in an elderly care domain (an anomaly is detected if actions' duration deviate from normal), and to the tasks of learning and making on-line classifications in series of activities related to home scenarios. S-HSMMs admit for dealing with some probabilistic restrictions over the duration of plans as well as the capability to detect anomalies. However, the model does not support concurrent activities, nor does it addresses matching with multi-feature observations.

Many probabilistic inference models were designed to be applied to activity recognition (e.g., [FHL⁺03, CG93, PW00]), these works emphasise the necessity for methods which improve the probabilistic recognition models for activities and goals, however, they do not model state duration in their computations. Fox et al. [FHL⁺03] review Bayesian filter implementations and techniques while concentrating on location estimation tasks. These techniques are important for circumstances common to any real-world application (e.g., situations in which location sensors provide errors or imperfect measurements) by representing uncertainty at different levels of abstraction. Bayesian filters do not use statistics on state duration in their computations and rely on the Markov assumption. Bayesian-networks models by Charniak and Goldman [CG93] and parsing of *Probabilistic State-Dependent Grammars* (PSDG) by Pynadath and Wellman [PW00] are statistical models for recognition, in order to deal with uncertainty in recognition tasks. In both of these, there is no explicit treatment of the importance of the amount of time spent in a certain state.

A probabilistic plan recognition algorithm for multi-agent overhearing is proposed in YOYO* [KPT02], where information about the average duration of plan steps are included in the plan library. This information is used to estimate the likelihood of an agent changing a step to another without being observed to do so (i.e., the system uses average duration of plan-step as an input, to estimate the duration of unobserved plan-steps). They utilised a recognition model that is based on hierarchical non-hidden semi-Markov models. This model presented unsatisfactory results in correctly predicting the joint states of multi-agent teams, due to the unmodeled variance of real duration.

This algorithm (YOYO*) does not address matching multi-feature observations neither allow for concurrent plans.

A hybrid symbolic probabilistic plan recognition system is presented by Geib and Harp [GH04], which allows for some partial observability as well as interleaved plans. However, the system does not address efficient matching of multi-feature observations, nor allows for taking duration into account. Tapia [Mun03] introduces a system for recognising activities in the home setting that uses machine learning algorithms, a set of small and simple state-change sensors, and electronic experience sampling. This work developed an algorithm for activity recognition that extends the naive Bayes classifier to incorporate low-order temporal relationships since temporal information is an important component of activities, however, this approach does not support concurrent plans neither multi-valued features.

In Avrahami-Zilberbrand et al. [AZKZ05] some modifications are presented in SBR algorithms allowing to handle intermittent failures in observations, lossy observations, and plans with duration. However, while dealing with plan execution duration constraints the algorithm cannot recognise interleaved plans. Although this approach can deal lossy observation and interrupted plan, added these recognition capabilities can significantly influence runtime, and the number of possible hypotheses (ambiguity). In Mirsky et al. [MG16] present an approach for online plan recognition called SLIM (Semi-Lazy Inference Mechanism), which uses parsing processes to commit only to the minimum necessary actions in real-time, but still provide complete hypotheses. Mirsky also present an approach that extends the Probabilistic Hostile Agent Task Tracker (PHATT) algorithm [GG09] called Cumulative Recognition of Activities and Decreasing Load of Explanations (CRADLE) [MGS17]. This online plan recognition algorithm is able to capture partial plans and exogenous actions by using domain-independent heuristics to prune the space of possible explanations and updating arguments in the plan to remain the explanations consistent with new observations. Concerning in the problem of how to disambiguate multiple hypotheses Mirsky et al. [MSGK16] present the Sequential Plan Recognition Process (SPRP) that disambiguate hypotheses by querying the observed agent about whether a candidate plan in one of the hypotheses matches its intentions, so this approach aims to reduce the number of hypotheses using a minimal number of queries.

Fagundes et al. [FMBV14] face the problem of deciding when to interact with the observed agent to determine his/her plan under execution by extending the SBR algorithm with a method named Expected Recognition Time (ERT) that receives as input a sequence of observations, and estimates the expected time to recognise a plan from multiple hypotheses obtained from the SBR. Moreover, this work developed a method named Plan Selection Counter (PSC) to save data about the number of times that a plan is selected, based in the context of execution. This approach has a limitation because it

does not allow for concurrent execution; that is, at each time step, only one plan step can be identified as an activity involving the agent. Although not considering concurrent plans, this approach demonstrated that SBR can be extended to allow an estimate of the time needed to recognise a plan by keeping a record of plan steps executed in a given context at a given moment. So, this work is the basis for our idea of the temporal plan recognition algorithm that is intended to keep information about the duration of each plan step linked with a plan being performed by the observed agent, storing, in addition, how many times this information was obtained in order to determine data accuracy. Table 7.1 shows the characteristics present in the main related work that are present in our approach.

Table 7.1: Characteristics of the main related work

	Our Approach	Helaoui [HNS11]	Fagundes [FMBV14]	Duong [DBPV05]	Tapia [Mun03]	Mirsky [MSGK16]	Dorit [AZKZ05]
Symbolic approach	✓		✓			✓	✓
Probabilistic approach		✓		✓	✓		
Concurrent activities	✓	✓				✓	✓
Interleaved activities	✓	✓				✓	✓
Activity duration	✓	✓	✓	✓	✓		
Multi-feature observations	✓		✓			✓	✓
Detect anomalies	✓		✓	✓			
Multivalued features	✓		✓			✓	✓
Lossy observations							✓
Learns patterns	✓	✓	✓	✓	✓		
Inference of start/end time	✓	✓			✓		
Dealing with ambiguity	✓		✓			✓	

8. CONCLUSION

In this work, we addressed one of the ultimate goals of Artificial Intelligence, which involves observing agent actions and environment changes aiming to predict agent's intentions and detect failures in plan execution. The process of inferring agent's plans from their observed actions is known as plan recognition. Most approaches to plan recognition are based on a knowledge base manually constructed, and represented as plan library for recognising activities and plans. Thus, first we developed a plan library generator, managed through a number of parameters to automatically create arbitrarily complex knowledge base, representing the sequence of activities that an observed agent can perform in the environment. In addition, we improved this generator to create temporal plan libraries that may store temporal information about activities, such as minimum and maximum duration of each plan step, besides time of day that a plan usually is performed.

We also developed the temporal input set generator to get observations from the temporal plan library. This method takes into account information about plan step duration to repeat, a determined number of times, the features associated with the activity in the set of observations simulating the activity duration. However, most plan recognition approaches cannot control duration bounds in plan steps nor lead to more complex forms of temporal plans, such as concurrent and interleaved plans, because they assume that a sequence of actions is wholly coherently performed towards a single goal and agents carries out only one activity at a time.

Consequently, there is a lack of systems to assist agents that perform multiple concurrent or interleaving activities able to automatic monitoring possible fails. Furthermore, most approaches cannot treat multi-signals from devices in realistic environments, because they assume only a binary signal representing interaction or no interaction with a device, besides agents interacting only with a single device at time. Nowadays, there are many powerful devices equipped with multiple sensors and functionalities that may be used to interact with an agent aiming to provide useful information about the environment and act as an assistant to help the agent to realise actions and plans.

In this work we developed a Temporal Plan Recognition approach based on the extended version of Symbolic Behaviour Recognition (SBR) responsible to return the set of candidate plans. We use the Plan Execution Controller (PEC) to filter these hypotheses aiming to eliminate ambiguity and check consistency in execution of each candidate plan. As observed in experiments using real datasets, precision can be hard to fulfil since the recogniser may return multiple candidates plans for a long time, especially in environments with many top plans and few features available. In this case, the high number of plans that start being recognised in current recognition episode but stop

being recognised in few next episodes are responsible for a permanent ambiguity in the set of hypotheses.

In systems developed to monitor and help an agent in completion of its Activities of Daily Living (ADL), ambiguity can make the application extremely intrusive when considering, for example, sending messages to the user whenever an abandoned plan is detected. To deal the problems of ambiguity in multiple hypotheses and detecting anomalies in a plan sequence, we developed an efficient algorithm to filter hypotheses by applying temporal and path length constraints using both the hierarchical organisation of activities and their expected time of day and duration acquired in previous episodes. To evaluate our approach, we analysed the results of the filter component present in PEC applied to hypothesis set returned by SBR in several experiments carried out both in real and generated datasets comparing the performance between SBR and PEC. As a result, in all experiments we note a great improvement in PEC precision compared to SBR, without a great decrease in recall.

Therefore, we developed an approach to extend Symbolic Behaviour Recognition (SBR) allowing an efficient control and disambiguation of multiple hypotheses consistent with the observations, besides dealing concurrent/interleaved plans and duration constraints in plan execution. Simulations in real and generated domains show a potential improvement in most evaluation metrics, especially in the precision measure. Besides that, our approach can detect anomalies in activities duration, since we can capture the normal patterns in duration spent for each activity, so it can be used to detect anomalies or possible failures in a particular plan sequence. Then, our approach is adequate for applications requiring temporal failure prediction, where the system must observe an agent and detect possible fails, acting to give support in time, or even before, the failure occurs.

Finally, as future work, we intend to improve the mechanism of learning the time of the day a plan is performed. Currently, when a plan is tagged as undefined hour, this hour can no longer be changed, that is, even this plan has been presenting a specific time of day, our approach is not longer able to learn this new hour. Furthermore, our current approach has a limitation in that we do not allow for lossy observations, but we intend to implement this technique in a new extension of SBR. Consequently, we also aim to conduct experiments to test the ability of our approach to deal with interleaved plans and lossy observation. Another point to consider is the question of how many timestamps can pass from the time an activity is interrupted, until returning to it. In general, we intend to integrate our approach into a larger application in order to perform experiments in more complex domains, within places in which multiple sensors, actuators and automation equipment coexist, providing several features multi-valued used to recognise the agent current plan(s) and checking its sequence of activities to identify possible failures.

REFERENCES

- [AA07] Armentano, M. G.; Amandi, A. "Plan recognition for interface agents", *Artificial Intelligence*, vol. 28-2, Aug 2007, pp. 131-162.
- [AFH89] Azarewicz, J.; Fala, G.; Heithecker, C. "Template-based multi-agent plan recognition for tactical situation assessment". In: Proceedings of the 5th Conference on Artificial Intelligence Applications, 1989, pp. 247-254.
- [AIM10] Atzori, L.; Iera, A.; Morabito, G. "The internet of things: a survey", *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 54-15, Oct 2010, pp. 2787-2805.
- [AM05] Abowd, G. D.; Mynatt, E. D. "Designing for the human experience in smart environments". In: *Smart Environments: Technologies, Protocols, and Applications*, John Wiley & Sons, 2005, chap. 7, pp. 151-174.
- [AMC07] Akyildiz, I. F.; Melodia, T.; Chowdhury, K. R. "A survey on wireless multimedia sensor networks", *Computer Networks*, vol. 51-4, Mar 2007, pp. 921-960.
- [Ami12] Amir, E. A. H. E. "On uses of mean absolute deviation: decomposition, skewness and correlation coefficients", *METRON*, vol. 70-2-3, Aug 2012, pp. 145-164.
- [AN06] Augusto, J. C.; Nugent, C. D. "Smart homes can be smarter". In: *Designing Smart Homes: The Role Of Artificial Intelligence*, Springer, 2006, chap. 1, pp. 1-15.
- [ASSC02] Akyildiz, I.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. "A survey on sensor networks", *IEEE Communications Magazine*, vol. 40-8, Aug 2002, pp. 102-114.
- [Avr09] Avrahami-Zilberbrand, D. "Efficient hybrid algorithms for plan recognition and detection of suspicious and anomalous behavior", Ph.D. Thesis, Bar Ilan University, 2009, 127p.
- [AZK05] Avrahami-Zilberbrand, D.; Kaminka, G. A. "Fast and complete symbolic plan recognition". In: Proceedings of the 19th International Joint Conference on Artificial Intelligence, 2005, pp. 653-658.
- [AZKZ05] Avrahami-Zilberbrand, D.; Kaminka, G. A.; Zarosim, H. "Fast and complete plan recognition: allowing for duration, interleaved execution, and lossy observations". In: Proceedings of the 19th International Joint Conference

on Artificial Intelligence Workshop on Modeling Others from Observations, 2005, pp. 1–7.

- [AZN98] Albrecht, D. W.; Zukerman, I.; Nicholson, A. E. “Bayesian models for keyhole plan recognition in an adventure game”, *User Modeling and User-Adapted Interaction*, vol. 8–1-2, Mar 1998, pp. 5–47.
- [BA05] Blaylock, N.; Allen, J. “Generating artificial corpora for plan recognition”. In: *User Modeling 2005*, Springer, 2005, chap. 21, pp. 179–188.
- [BBC⁺00] Baldi, P.; Brunak, S.; Chauvin, Y.; Andersen, C. A. F.; Nielsen, H. “Assessing the accuracy of prediction algorithms for classification: an overview”, *Bioinformatics*, vol. 16–5, May 2000, pp. 412–424.
- [BBF15] Benmansour, A.; Bouchachia, A.; Feham, M. “Human activity recognition in pervasive single resident smart homes: state of art”. In: Proceedings of the 12th International Symposium on Programming and Systems, 2015, pp. 1–9.
- [BFA00] Burstein, M. H.; Ferguson, G.; Allen, J. F. “Integrating agent-based mixed-initiative control with an existing multi-agent planning system”. In: Proceedings of the 4th International Conference on Multi-Agent Systems, 2000, pp. 389–390.
- [BKL10] Banerjee, B.; Kraemer, L.; Lyle, J. “Multi-agent plan recognition: formalization and algorithms”. In: Proceedings of the 24th Association for the Advancement of Artificial Intelligence, 2010, pp. 1059–1064.
- [BPV04] Bui, H. H.; Phung, D. Q.; Venkatesh, S. “Learning hierarchical hidden markov models with general state hierarchy”. In: Proceedings of the 19th National Conference on Artificial Intelligence, 2004, pp. 324–329.
- [Bro98] Brown, S. M. “A decision theoretic approach for interface agent development”, Master’s Thesis, Air Force Institute of Technology, 1998, 190p.
- [BST09] Baker, C.; Saxe, R.; Tenenbaum, J. “Action understanding as inverse planning”, *Cognition*, vol. 113–3, Dec 2009, pp. 329–349.
- [BVW02] Bui, H. H.; Venkatesh, S.; West, G. “Policy recognition in the abstract hidden markov model”, *Journal of Artificial Intelligence Research*, vol. 17–1, Dec 2002, pp. 451–499.
- [Car01] Carberry, S. “Techniques for plan recognition”, *User Modeling and User-Adapted Interaction*, vol. 11–2, Mar 2001, pp. 31–48.

- [CCAY13] Chernbumroong, S.; Cang, S.; Atkins, A.; Yu, H. "Elderly activities recognition and classification for applications in assisted living", *Expert Systems with Applications*, vol. 40–5, Apr 2013, pp. 1662–1674.
- [CCS⁺17] Calvaresi, D.; Cesarini, D.; Sernani, P.; Marinoni, M.; Dragoni, A. F.; Sturm, A. "Exploring the ambient assisted living domain: a systematic review", *Journal of Ambient Intelligence and Humanized Computing*, vol. 8–2, Apr 2017, pp. 239–257.
- [CG90] Charniak, E.; Goldman, R. P. "Plan recognition in stories and in life", *Machine Intelligence and Pattern Recognition*, vol. 10–1, Jun 1990, pp. 343–351.
- [CG93] Charniak, E.; Goldman, R. P. "A bayesian model of plan recognition", *Artificial Intelligence*, vol. 64–1, Nov 1993, pp. 53–79.
- [CK11] Chen, L.; Khalil, I. "Activity recognition: approaches, practices and trends". In: *Activity Recognition in Pervasive Intelligent Environments*, Atlantis Press, 2011, chap. 1, pp. 1–31.
- [CKS10] Cirillo, M.; Karlsson, L.; Saffiotti, A. "Human-aware task planning: an application to mobile robots", *ACM Transactions on Intelligent Systems and Technology*, vol. 1–2, Dec 2010, pp. 1–26.
- [DBPV05] Duong, T. V.; Bui, H. H.; Phung, D. Q.; Venkatesh, S. "Activity recognition and abnormality detection with the switching hidden semi-markov model". In: *Proceedings of the 12th IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 838–845.
- [DG06] Davis, J.; Goadrich, M. "The relationship between precision-recall and ROC curves". In: *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 233–240.
- [DMG⁺16] Dobre, C.; Mavromoustakis, C. X.; Garcia, N.; Goleva, R. I.; Mastorakis, G. "Ambient assisted living and enhanced living environments: principles, technologies and control". Butterworth-Heinemann, 2016, 552p.
- [DPBV09] Duong, T.; Phung, D.; Bui, H.; Venkatesh, S. "Efficient duration and hierarchical modeling for human activity recognition", *Artificial Intelligence*, vol. 173–8, May 2009, pp. 830–856.
- [EHN94a] Erol, K.; Hendler, J. A.; Nau, D. S. "HTN planning: complexity and expressivity". In: *Proceedings of the 12th National Conference on Artificial Intelligence*, 1994, pp. 1123–1128.

- [EHN94b] Erol, K.; Hendler, J. A.; Nau, D. S. "UMCP: a sound and complete procedure for hierarchical task-network planning". In: Proceedings of the 2nd International Conference on Artificial Intelligence Planning Systems, 1994, pp. 249–254.
- [Faw06] Fawcett, T. "An introduction to ROC analysis", *Pattern Recognition Letters*, vol. 27–8, Jun 2006, pp. 861–874.
- [FHB⁺08] Frade, F. D.; Hodgins, J. K.; Bargteil, A. W.; Artal, X. M.; Macey, J. C.; Castells, A. C. I.; Beltran, J. "Guide to the carnegie mellon university multimodal activity (CMU-MMAC) database", Technical Report CMU-RI-TR-08-22, Carnegie Mellon University, 2008, 17p.
- [FHL⁺03] Fox, D.; Hightower, J.; Liao, L.; Schulz, D.; Borriello, G. "Bayesian filtering for location estimation", *IEEE Pervasive Computing*, vol. 2–3, Jul 2003, pp. 24–33.
- [FHM⁺15] Farias, G. P.; Hilgert, L. W.; Meneguzzi, F. R.; Vieira, R.; Bordini, R. H. "Automatic generation of plan libraries for plan recognition performance evaluation". In: Proceedings of the 14th IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2015, pp. 129–132.
- [FL03] Fox, M.; Long, D. "PDDL2.1: an extension to PDDL for expressing temporal planning domains", *Journal of Artificial Intelligence Research*, vol. 20–1, Dec 2003, pp. 61–124.
- [FMBV14] Fagundes, M.; Meneguzzi, F.; Bordini, R. H.; Vieira, R. "Dealing with ambiguity in plan recognition under time constraints". In: Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems, 2014, pp. 389–396.
- [FPH⁺16] Farias, G.; Pereira, R. F.; Hilgert, L.; Meneguzzi, F.; Vieira, R.; Bordini, R. H. "Failure prediction based on monitoring sequences of actions and action duration". In: Proceedings of the 10th Software Agents, Environments and Applications School, 2016, pp. 70–81.
- [FPH⁺17] Farias, G.; Pereira, R.; Hilgert, L.; Meneguzzi, F.; Vieira, R.; Bordini, R. "Predicting plan failure by monitoring action sequences and duration", *Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 6–2, Apr 2017, pp. 71–84.
- [FXL⁺17] Fan, X.; Xie, Q.; Li, X.; Huang, H.; Wang, J.; Chen, S.; Xie, C.; Chen, J. "Activity recognition as a service for smart home: ambient assisted living

application via sensing home”. In: Proceedings of the 6th IEEE International Conference on AI and Mobile Services, 2017, pp. 54–61.

- [Gei02] Geib, C. W. “Problems with intent recognition for elder care”. In: Proceedings of the 18th National Conference on Artificial Intelligence Workshop Automation as Caregiver: the Role of Intelligent Technology in Elder Care, 2002, pp. 13–17.
- [Gei04] Geib, C. W. “Assessing the complexity of plan recognition”. In: Proceedings of the 19th National Conference on Artificial Intelligence, 2004, pp. 507–512.
- [GG01a] Geib, C. W.; Goldman, R. P. “Plan recognition in intrusion detection systems”. In: Proceedings of the 2nd DARPA Information Survivability Conference and Exposition, 2001, pp. 46–55.
- [GG01b] Geib, C. W.; Goldman, R. P. “Probabilistic plan recognition for hostile agents”. In: Proceedings of the 14th International Florida Artificial Intelligence Research Society Conference, 2001, pp. 580–584.
- [GG03] Geib, C. W.; Goldman, R. P. “Recognizing plan/goal abandonment”. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence, 2003, pp. 1515–1517.
- [GG09] Geib, C. W.; Goldman, R. P. “A probabilistic plan recognition algorithm based on plan tree grammars”, *Artificial Intelligence*, vol. 173–11, Jul 2009, pp. 1101–1132.
- [GGM99] Goldman, R. P.; Geib, C. W.; Miller, C. A. “A new model of plan recognition”. In: Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence, 1999, pp. 245–254.
- [GH04] Geib, C.; Harp, S. “Empirical analysis of a probabilistic task tracking algorithm”. In: Proceedings of the 3rd International Joint Conference on Autonomous Agent Systems Workshop on Modeling Other Agents from Observations, 2004, pp. 1–7.
- [GHK99] Grosz, B. J.; Hunsberger, L.; Kraus, S. “Planning and acting together”, *AI Magazine*, vol. 20–4, Dec 1999, pp. 23–34.
- [GIMA10] Giusto, D.; Iera, A.; Morabito, G.; Atzori, L. “The internet of things 20th tyrrhenian workshop on digital communications”. Springer, 2010, 442p.
- [GK95] Greer, J. E.; Koehn, G. M. “The peculiarities of plan recognition for intelligent tutoring systems”. In: Proceedings of the 14th International Joint

Conference on Artificial Intelligence Workshop on the Next Generation of Plan Recognition Systems, 1995, pp. 54–59.

- [GS90] Grosz, B. J.; Sidner, C. L. “Plans for discourse”. In: *Intentions in Communication*, MIT Press, 1990, chap. 20, pp. 417–444.
- [HBH⁺98] Horvitz, E. J.; Breese, J. S.; Heckerman, D.; Hovel, D.; Rommelse, K. “The Lumière project: Bayesian user modeling for inferring the goals and needs of software users”. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 1998, pp. 256–265.
- [HD93] Huber, M.; Durfee, E. H. “Observational uncertainty in plan recognition among interacting robots”. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence Workshop on Dynamically Interacting Robots, 1993, pp. 68–75.
- [HHPZ⁺08] Hao Hu, D.; Pan, S. J.; Zheng, V. W.; Liu, N. N.; Yang, Q. “Real world activity recognition with multiple goals”. In: Proceedings of the 10th International Conference on Ubiquitous Computing, 2008, pp. 30–39.
- [HLR⁺13] Hong, K.; Lillethun, D.; Ramachandran, U.; Ottenwalder, B.; Koldehofe, B. “Mobile fog: a programming model for large-scale applications on the internet of things”. In: Proceedings of the 2nd ACM Special Interest Group on Data Communication Workshop on Mobile Cloud Computing, 2013, pp. 15–20.
- [HN03] Hongeng, S.; Nevatia, R. “Large-scale event detection using semi-hidden markov models”. In: Proceedings of the 9th IEEE International Conference on Computer Vision, 2003, pp. 1455–1462.
- [HNS11] Helaoui, R.; Niepert, M.; Stuckenschmidt, H. “Recognizing interleaved and concurrent activities: a statistical-relational approach”. In: Proceedings of the 9th IEEE International Conference on Pervasive Computing and Communications, 2011, pp. 1–9.
- [Hon01] Hong, J. “Goal recognition through goal graph analysis”, *Journal of Artificial Intelligence Research*, vol. 15–1, Jul 2001, pp. 1–30.
- [HZY⁺09] Hu, D. H.; Zhang, X. X.; Yin, J.; Zheng, V. W.; Yang, Q. “Abnormal activity recognition based on HDP-HMM models”. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, 2009, pp. 1715–1720.
- [IB99] Intille, S. S.; Bobick, A. F. “A framework for recognizing multi-agent action from visual evidence”. In: Proceedings of the 16th National Conference on

Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence, 1999, pp. 518–525.

- [KA86] Kautz, H. A.; Allen, J. F. “Generalized plan recognition”. In: Proceedings of the 5th National Conference on Artificial Intelligence, 1986, pp. 32–37.
- [KA14] Krzywinski, M.; Altman, N. “Points of significance: visualizing samples with box plots”, *Nature Methods*, vol. 11–2, Feb 2014, pp. 119–120.
- [KBdB⁺13] Koedinger, K. R.; Brunskill, E.; de Baker, R. S. J.; McLaughlin, E. A.; Stamper, J. C. “New potentials for data-driven intelligent tutoring system development and optimization”, *AI Magazine*, vol. 34–3, Sep 2013, pp. 27–41.
- [KP14] Krishnan, R. H.; Pugazhenti, S. “Mobility assistive devices and self-transfer robotic systems for elderly, a review”, *Intelligent Service Robotics*, vol. 7–1, Jan 2014, pp. 37–49.
- [KPT02] Kaminka, G. A.; Pynadath, D. V.; Tambe, M. “Monitoring teams by overhearing: a multi-agent plan recognition approach”, *Journal of Artificial Intelligence Research*, vol. 17–1, Jul 2002, pp. 83–135.
- [KRS15] Kilger, C.; Reuter, B.; Stadtler, H. “Collaborative planning”. In: *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies*, Springer, 2015, chap. 14, pp. 257–277.
- [KRSA13] Kahrobaee, S.; Rajabzadeh, R. A.; Soh, L. K.; Asgarpoor, S. “A multiagent modeling and investigation of smart homes with power generation, storage, and trading features”, *IEEE Transactions on Smart Grid*, vol. 4–2, Jun 2013, pp. 659–668.
- [KT00] Kaminka, G. A.; Tambe, M. “Robust agent teams via socially-attentive monitoring”, *Journal of Artificial Intelligence Research*, vol. 12–1, Mar 2000, pp. 105–147.
- [LCY⁺15] Liu, Y.; Cheng, J.; Yan, C.; Wu, X.; Chen, F. “Research on the Matthews correlation coefficients metrics of personalized recommendation algorithm evaluation”, *International Journal of Hybrid Information Technology*, vol. 8–1, Jan 2015, pp. 163–172.
- [LFH⁺03] Liao, L.; Fox, D.; Hightower, J.; Kautz, H.; Schulz, D. “Voronoi tracking: location estimation using sparse and noisy sensor data”. In: Proceedings of the 16th IEEE International Conference on Intelligent Robots and Systems, 2003, pp. 723–728.

- [LRS99] Lesh, N.; Rich, C.; Sidner, C. L. "Using plan recognition in human-computer collaboration". In: Proceedings of the 7th International Conference on User Modeling, 1999, pp. 23–32.
- [LWY14] Li, L.; Wen, D.; Yao, D. "A survey of traffic control with vehicular communications", *IEEE Transactions on Intelligent Transportation Systems*, vol. 15–1, Feb 2014, pp. 425–432.
- [LXAR13] Liu, R.; Xie, X.; Augusto, V.; Rodriguez, C. "Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care", *European Journal of Operational Research*, vol. 230–3, Nov 2013, pp. 475–486.
- [Mat75] Matthews, B. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme", *Biochimica et Biophysica Acta - Protein Structure*, vol. 405–2, Oct 1975, pp. 442–451.
- [McC86] McCarthy, J. "Applications of circumscription to formalizing common-sense knowledge", *Artificial Intelligence*, vol. 28–1, Feb 1986, pp. 89–116.
- [MFRR15] Monekosso, D.; Florez-Revuelta, F.; Remagnino, P. "Ambient assisted living", *IEEE Intelligent Systems*, vol. 30–4, Jul 2015, pp. 2–6.
- [MG04] Mao, W.; Gratch, J. "A utility-based approach to intention recognition". In: Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, 2004, pp. 1–6.
- [MG16] Mirsky, R.; Gal, Y. "SLIM: semi-lazy inference mechanism for plan recognition". In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, 2016, pp. 394–400.
- [MGS17] Mirsky, R.; Gal, Y. K.; Shieber, S. M. "CRADLE: an online plan recognition algorithm for exploratory domains", *ACM Transactions on Intelligent Systems and Technology*, vol. 8–3, Apr 2017, pp. 1–22.
- [MHKF09] Marhasev, E.; Hadad, M.; Kaminka, G. A.; Feintuch, U. "The use of hidden semi-markov models in clinical diagnosis maze tasks", *Intelligent Data Analysis*, vol. 13–6, Dec 2009, pp. 943–967.
- [MPPS17] Manca, M.; Parvin, P.; Paternò, F.; Santoro, C. "Detecting anomalous elderly behaviour in ambient assisted living". In: Proceedings of the 9th ACM Special Interest Group on Computer-Human Interaction Symposium on Engineering Interactive Computing Systems, 2017, pp. 63–68.

- [MSGK16] Mirsky, R.; Stern, R.; Gal, Y. K.; Kalech, M. "Sequential plan recognition". In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, 2016, pp. 401–407.
- [MTL78] McGill, R.; Tukey, J. W.; Larsen, W. A. "Variations of box plots", *The American Statistician*, vol. 32–1, Feb 1978, pp. 12–16.
- [Mun03] Munguia Tapia, E. "Activity recognition in the home setting using simple and ubiquitous sensors", Ph.D. Thesis, Massachusetts Institute of Technology, 2003, 136p.
- [MW08] Mausam; Weld, D. S. "Planning with durative actions in stochastic domains", *Journal of Artificial Intelligence Research*, vol. 31–1, Jan 2008, pp. 33–82.
- [OMSN13] Oh, J.; Meneguzzi, F.; Sycara, K. P.; Norman, T. J. "Prognostic normative reasoning", *Engineering Applications of Artificial Intelligence*, vol. 26–2, Feb 2013, pp. 863–872.
- [OnB15] Ontañón, S.; Buro, M. "Adversarial hierarchical-task network planning for complex real-time games". In: Proceedings of the 24th International Conference on Artificial Intelligence, 2015, pp. 1652–1658.
- [OnSU⁺13] Ontañón, S.; Synnaeve, G.; Uriarte, A.; Richoux, F.; Churchill, D.; Preuss, M. "A survey of real-time strategy game AI research and competition in starcraft", *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 5–4, Dec 2013, pp. 293–311.
- [PBC⁺03] Pollack, M. E.; Brown, L.; Colbry, D.; McCarthy, C. E.; Orosz, C.; Peintner, B.; Ramakrishnan, S.; Tsamardinos, I. "Autominder: an intelligent cognitive orthotic system for people with memory impairment", *Robotics and Autonomous Systems*, vol. 44–4, Sep 2003, pp. 273–282.
- [Pea88] Pearl, J. "Probabilistic reasoning in intelligent systems: networks of plausible inference". Morgan Kaufmann, 1988, 552p.
- [PFS⁺15] Panisson, A. R.; Freitas, A.; Schmidt, D.; Hilgert, L.; Meneguzzi, F.; Vieira, R.; Bordini, R. H. "Arguing about task reallocation using ontological information in multi-agent systems". In: Proceedings of the 12th International Workshop on Argumentation in Multiagent Systems, 2015, pp. 1–16.
- [Pow11] Powers, D. M. W. "Evaluation: from precision, recall and f-measure to ROC, informedness, markedness & correlation", *Journal of Machine Learning Technologies*, vol. 2–1, Feb 2011, pp. 37–63.

- [PW95] Pynadath, D. V.; Wellman, M. P. "Accounting for context in plan recognition, with application to traffic monitoring". In: Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence, 1995, pp. 472–481.
- [PW00] Pynadath, D. V.; Wellman, M. P. "Probabilistic state-dependent grammars for plan recognition". In: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, 2000, pp. 507–514.
- [RA15] Rautaray, S. S.; Agrawal, A. "Vision based hand gesture recognition for human computer interaction: a survey", *Artificial Intelligence Review*, vol. 43–1, Jan 2015, pp. 1–54.
- [RBBG10] Roy, P. C.; Bouchard, B.; Bouzouane, A.; Giroux, S. "Combining pervasive computing with activity recognition and learning". In: *Web Intelligence and Intelligent Agents*, InTech, 2010, chap. 22, pp. 447–462.
- [RC09] Rashidi, P.; Cook, D. "Keeping the resident in the loop: adapting the smart home to the user", *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 39–5, Sep 2009, pp. 949–959.
- [RD06] Richardson, M.; Domingos, P. "Markov logic networks", *Machine Learning*, vol. 62–2, Feb 2006, pp. 107–136.
- [RG09] Ramírez, M.; Geffner, H. "Plan recognition as planning". In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, 2009, pp. 1778–1783.
- [RS97] Rich, C.; Sidner, C. L. "COLLAGEN: when agents collaborate with people". In: Proceedings of the 1st International Conference on Autonomous Agents, 1997, pp. 284–291.
- [SA77] Schank, R. C.; Abelson, R. P. "Scripts, plans, goals and understanding : an inquiry into human knowledge structures". Lawrence Erlbaum Associates, 1977, 256p.
- [Sat01] Satyanarayanan, M. "Pervasive computing: vision and challenges", *IEEE Personal Communications*, vol. 8–4, Aug 2001, pp. 10–17.
- [SGB⁺14] Sukthankar, G.; Geib, C.; Bui, H. H.; Pynadath, D.; Goldman, R. P. "Plan, activity, and intent recognition". Elsevier, 2014, 424p.
- [SRS⁺16] Santos, J. a.; Rodrigues, J. J. P. C.; Silva, B. M. C.; Casal, J. a.; Saleem, K.; Denisov, V. "An IoT-based mobile gateway for intelligent personal assistants on mobile health environments", *Journal of Network and Computer Applications*, vol. 71–1, Mar 2016, pp. 194–204.

- [SS08] Sukthankar, G.; Sycara, K. P. "Robust and efficient plan recognition for dynamic multi-agent teams". In: Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems, 2008, pp. 1383–1388.
- [SS11] Sukthankar, G.; Sycara, K. "Activity recognition for dynamic multi-agent teams", *ACM Transactions on Intelligent Systems and Technology*, vol. 3–1, Oct 2011, pp. 1–24.
- [SSG78] Schmidt, C. F.; Sridharan, N. S.; Goodson, J. L. "The plan recognition problem: an intersection of psychology and artificial intelligence", *Artificial Intelligence*, vol. 11–2, Aug 1978, pp. 45–83.
- [Syc02] Sycara, K. "Integrating agents into human teams". In: Proceedings of the 46th Human Factors and Ergonomics Society Annual Meeting, 2002, pp. 413–417.
- [TIL04] Tapia, E. M.; Intille, S. S.; Larson, K. "Activity recognition in the home using simple and ubiquitous sensors". In: Proceedings of the 2nd International Conference of Pervasive Computing, 2004, pp. 158–175.
- [TMIL04] Tapia, E. M.; Marmasse, N.; Intille, S. S.; Larson, K. "MITes: wireless portable sensors for studying behavior". In: Proceedings of the 6th ACM International Joint Conference on Pervasive and Ubiquitous Computing: Extended Abstracts Ubiquitous Computing, 2004, pp. 1–2.
- [UI00] Ullmer, B.; Ishii, H. "Emerging frameworks for tangible user interfaces", *IBM Systems Journal*, vol. 39–4, Jul 2000, pp. 915–931.
- [VA13] Vishwakarma, S.; Agrawal, A. "A survey on activity recognition and behavior understanding in video surveillance", *The Visual Computer*, vol. 29–10, Oct 2013, pp. 983–1009.
- [VVL07] Vail, D. L.; Veloso, M. M.; Lafferty, J. D. "Conditional random fields for activity recognition". In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, 2007, pp. 1–8.
- [WADX15] Whitmore, A.; Agarwal, A.; Da Xu, L. "The internet of things – a survey of topics and trends", *Information Systems Frontiers*, vol. 17–2, Apr 2015, pp. 261–274.
- [Wae96] Waern, A. "Recognising human plans: issues for plan recognition in human-computer interaction", Ph.D. Thesis, Royal Institute of Technology, 1996, 216p.

- [WHHB15] Wilson, C.; Hargreaves, T.; Hauxwell-Baldwin, R. "Smart homes and their users: a systematic analysis and key challenges", *Personal and Ubiquitous Computing*, vol. 19-2, Feb 2015, pp. 463-476.
- [WKL⁺08] Welbourne, E.; Khoussainova, N.; Letchner, J.; Li, Y.; Balazinska, M.; Borriello, G.; Suciu, D. "Cascadia: a system for specifying, detecting, and managing RFID events". In: Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services, 2008, pp. 281-294.
- [Yan09] Yang, Q. "Activity recognition: linking low-level sensors to high-level intelligence". In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, 2009, pp. 20-25.
- [YHY09] Yin, J.; Hu, D. H.; Yang, Q. "Spatio-temporal event detection using dynamic conditional random fields". In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, 2009, pp. 1321-1326.
- [ZL11] Zhuo, H. H.; Li, L. "Multi-agent plan recognition with partial team traces and plan libraries". In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, 2011, pp. 484-489.

APPENDIX A – SUBJECT ONE – TABLES AND GRAPHS

Table A.1: PEC, subject one, complete dataset, most plans with undefined hour

\mathcal{P}	δ	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.617	0.206	0.732	0.203	0.646	0.170	0.957	0.021	0.033	0.020	1360519
	2	0.730	0.237	0.698	0.232	0.701	0.184	0.967	0.020	0.020	0.018	1365184
	3	0.867	0.176	0.663	0.254	0.762	0.167	0.977	0.015	0.008	0.010	1372887
2	1	0.576	0.164	0.721	0.153	0.623	0.131	0.921	0.032	0.063	0.032	1728867
	2	0.678	0.192	0.690	0.170	0.663	0.148	0.939	0.028	0.041	0.027	1748420
	3	0.816	0.182	0.672	0.175	0.719	0.150	0.957	0.023	0.019	0.019	1701167
3	1	0.557	0.135	0.694	0.136	0.605	0.113	0.888	0.039	0.091	0.040	2028554
	2	0.661	0.159	0.664	0.140	0.648	0.122	0.912	0.034	0.060	0.033	1996923
	3	0.770	0.166	0.621	0.140	0.671	0.121	0.930	0.027	0.034	0.025	1994874

Table A.2: SBR, subject one, complete dataset, most plans with undefined hour

\mathcal{P}	δ	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.232	0.106	1.000	0.000	0.355	0.122	0.742	0.079	0.270	0.083	114232
	2	0.235	0.108	1.000	0.000	0.358	0.124	0.740	0.079	0.273	0.083	113854
	3	0.237	0.107	1.000	0.000	0.361	0.124	0.743	0.079	0.269	0.082	116951
2	1	0.233	0.058	1.000	0.000	0.372	0.073	0.684	0.069	0.343	0.075	222852
	2	0.236	0.059	1.000	0.000	0.375	0.074	0.686	0.067	0.341	0.074	228616
	3	0.240	0.063	1.000	0.000	0.379	0.079	0.687	0.067	0.340	0.073	223573
3	1	0.261	0.048	1.000	0.000	0.410	0.059	0.649	0.063	0.396	0.072	349005
	2	0.264	0.049	1.000	0.000	0.414	0.060	0.654	0.063	0.391	0.073	345103
	3	0.261	0.047	1.000	0.000	0.410	0.059	0.650	0.060	0.395	0.070	342255

Table A.3: PEC, subject one, complete dataset, all plans with defined hour

\mathcal{P}	δ	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.949	0.076	0.941	0.080	0.937	0.076	0.993	0.008	0.005	0.007	1436552
	2	0.959	0.065	0.905	0.127	0.939	0.074	0.992	0.009	0.004	0.006	1228737
	3	0.971	0.051	0.879	0.156	0.928	0.089	0.992	0.010	0.002	0.004	1231246
2	1	0.911	0.108	0.778	0.158	0.821	0.125	0.974	0.018	0.010	0.012	1572001
	2	0.913	0.115	0.758	0.156	0.812	0.130	0.973	0.018	0.009	0.012	1591897
	3	0.944	0.085	0.704	0.185	0.781	0.147	0.972	0.017	0.006	0.009	1538831
3	1	0.878	0.133	0.748	0.159	0.788	0.129	0.966	0.022	0.015	0.016	1619257
	2	0.902	0.121	0.727	0.162	0.785	0.130	0.967	0.020	0.012	0.014	1642936
	3	0.922	0.108	0.694	0.189	0.765	0.150	0.967	0.019	0.008	0.011	1725875

Table A.4: SBR, subject one, complete dataset, all plans with defined hour

\mathcal{P}	δ	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.247	0.113	1.000	0.000	0.373	0.129	0.754	0.074	0.258	0.077	140279
	2	0.233	0.101	1.000	0.000	0.358	0.119	0.744	0.079	0.268	0.082	116239
	3	0.237	0.107	1.000	0.000	0.362	0.123	0.744	0.075	0.268	0.078	106656
2	1	0.222	0.051	1.000	0.000	0.358	0.066	0.689	0.070	0.337	0.078	205344
	2	0.217	0.049	1.000	0.000	0.352	0.064	0.681	0.067	0.345	0.074	204669
	3	0.220	0.046	1.000	0.000	0.357	0.061	0.686	0.068	0.340	0.076	198779
3	1	0.225	0.053	1.000	0.000	0.362	0.068	0.676	0.072	0.353	0.083	223215
	2	0.227	0.052	1.000	0.000	0.365	0.068	0.678	0.071	0.352	0.081	245695
	3	0.221	0.051	1.000	0.000	0.357	0.067	0.672	0.070	0.358	0.081	256180

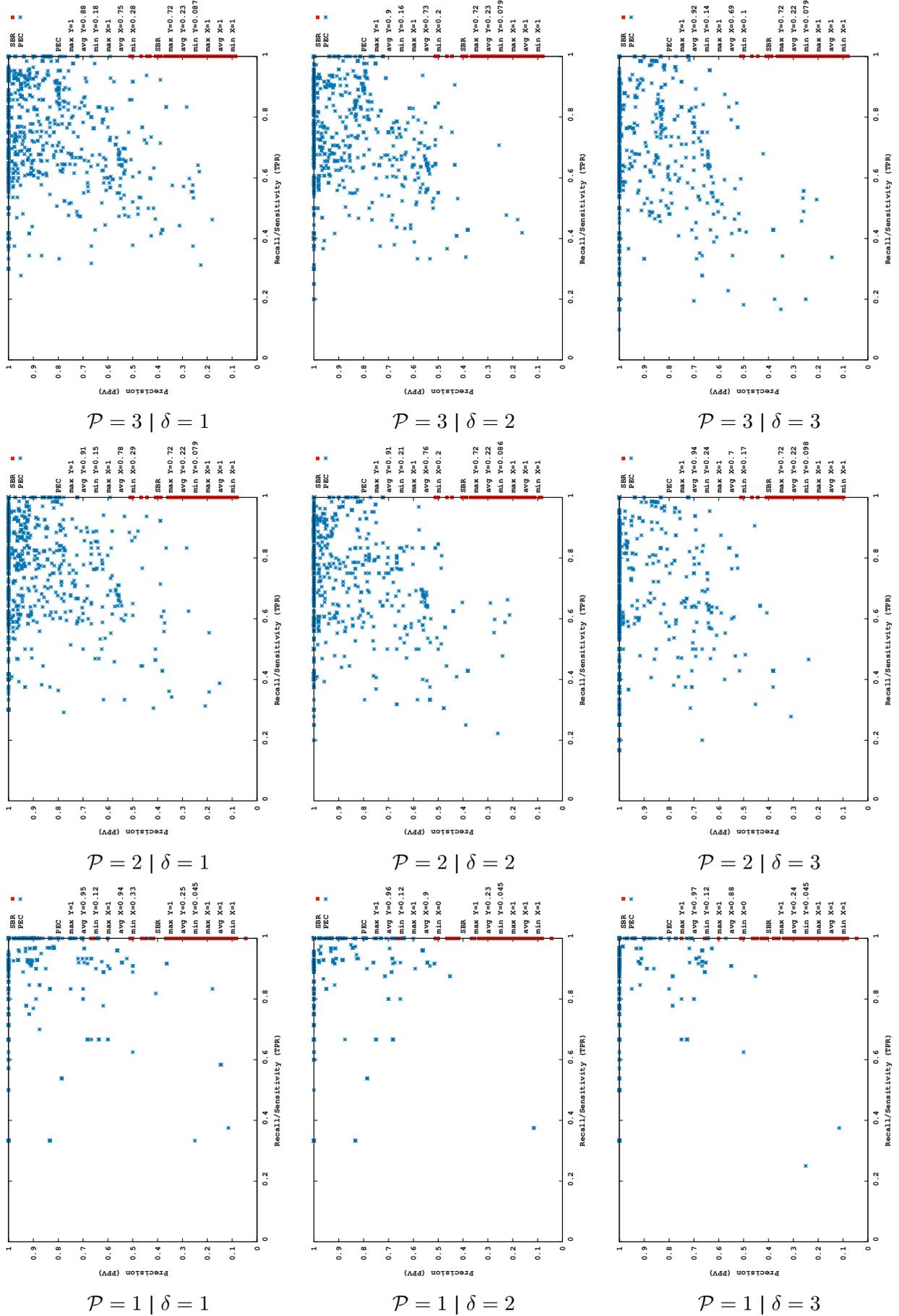


Figure A.2: PR space graphs switching parallel planes (\mathcal{P}) and path length (δ) for subject one using complete dataset where all plans have defined hour.

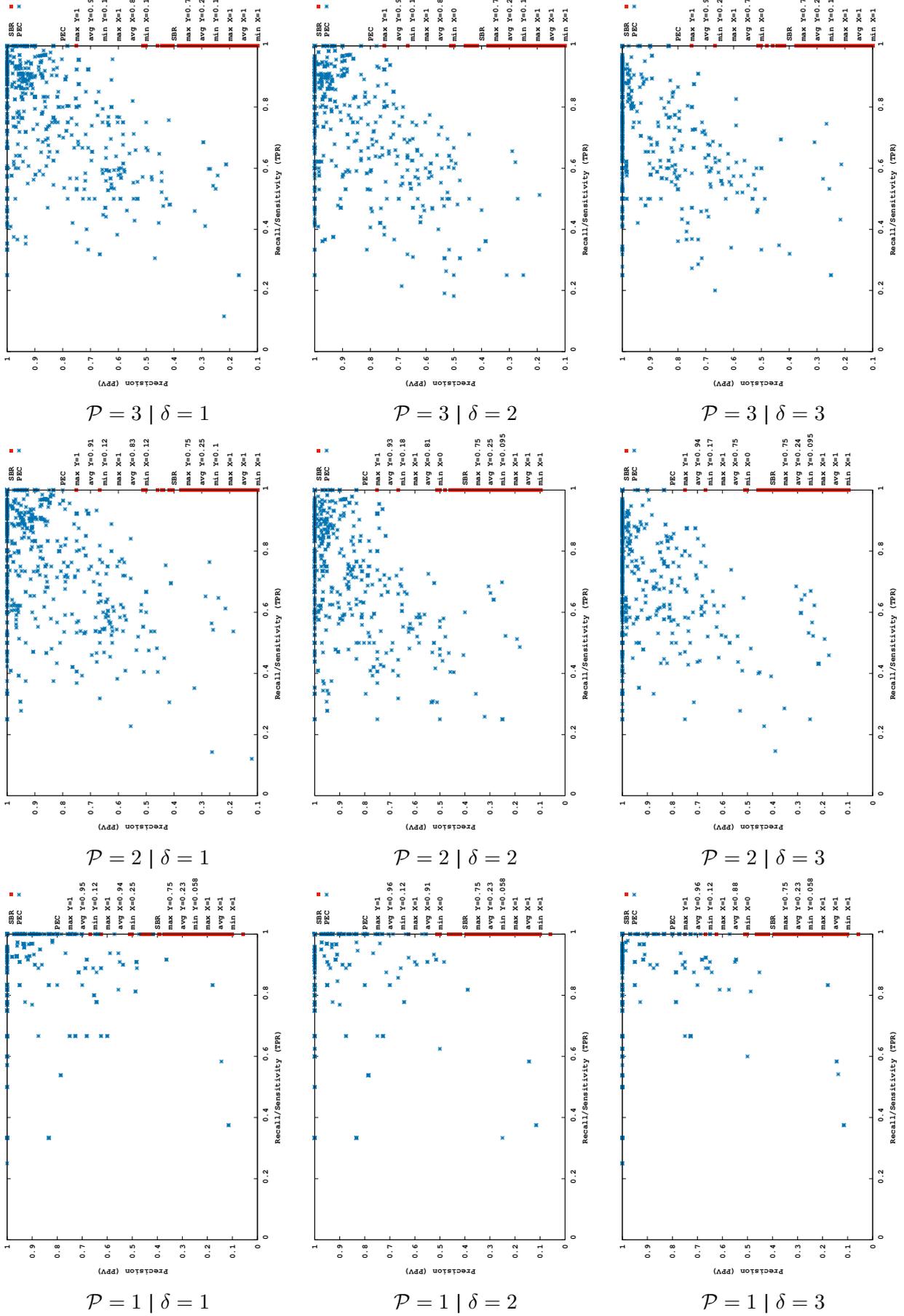


Figure A.4: PR space graphs switching parallel planes (\mathcal{P}) and path length (δ) for subject one using partial dataset where all plans have defined hour.

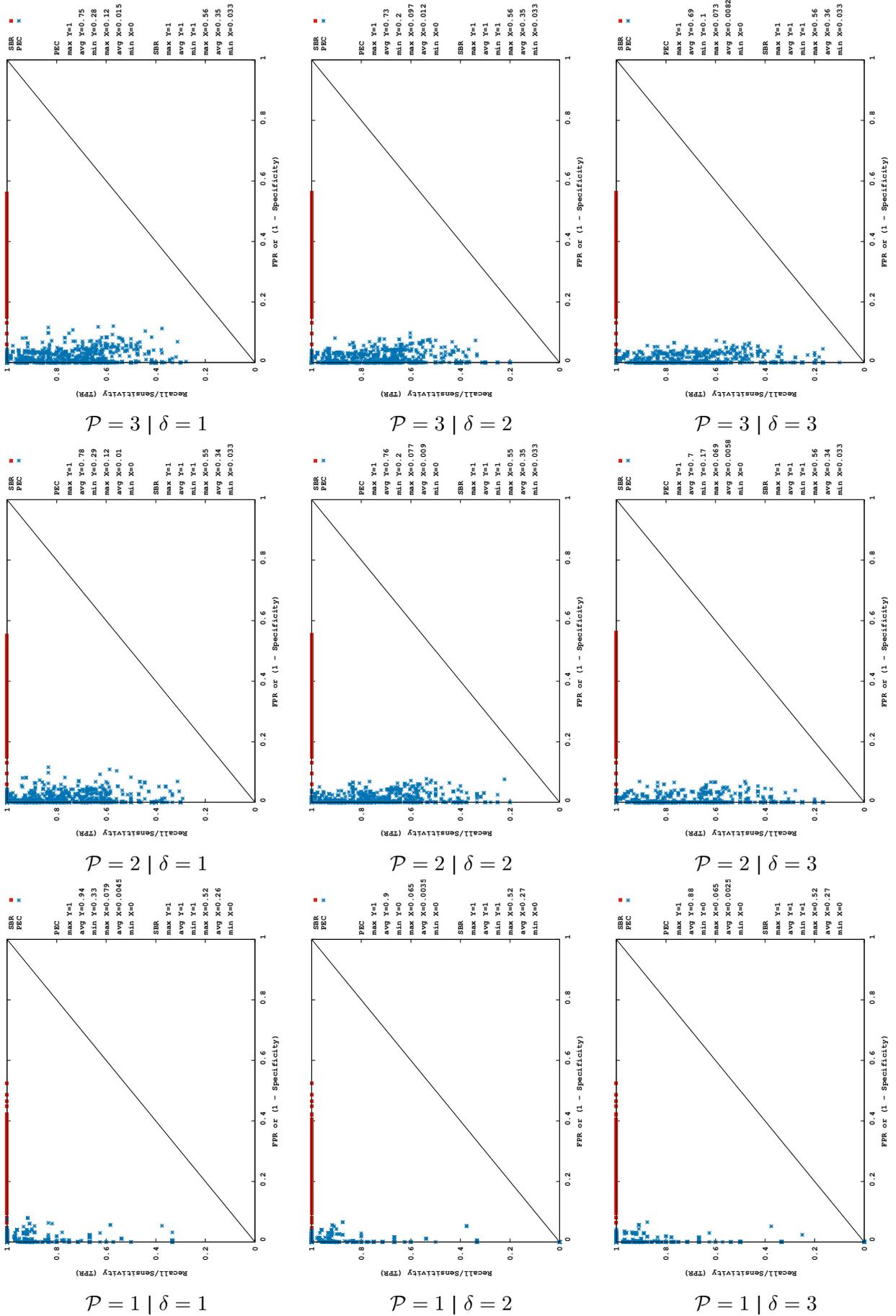


Figure A.6: ROC space graphs switching parallel plans (\mathcal{P}) and path length (δ) for subject one using complete dataset where all plans have defined hour.

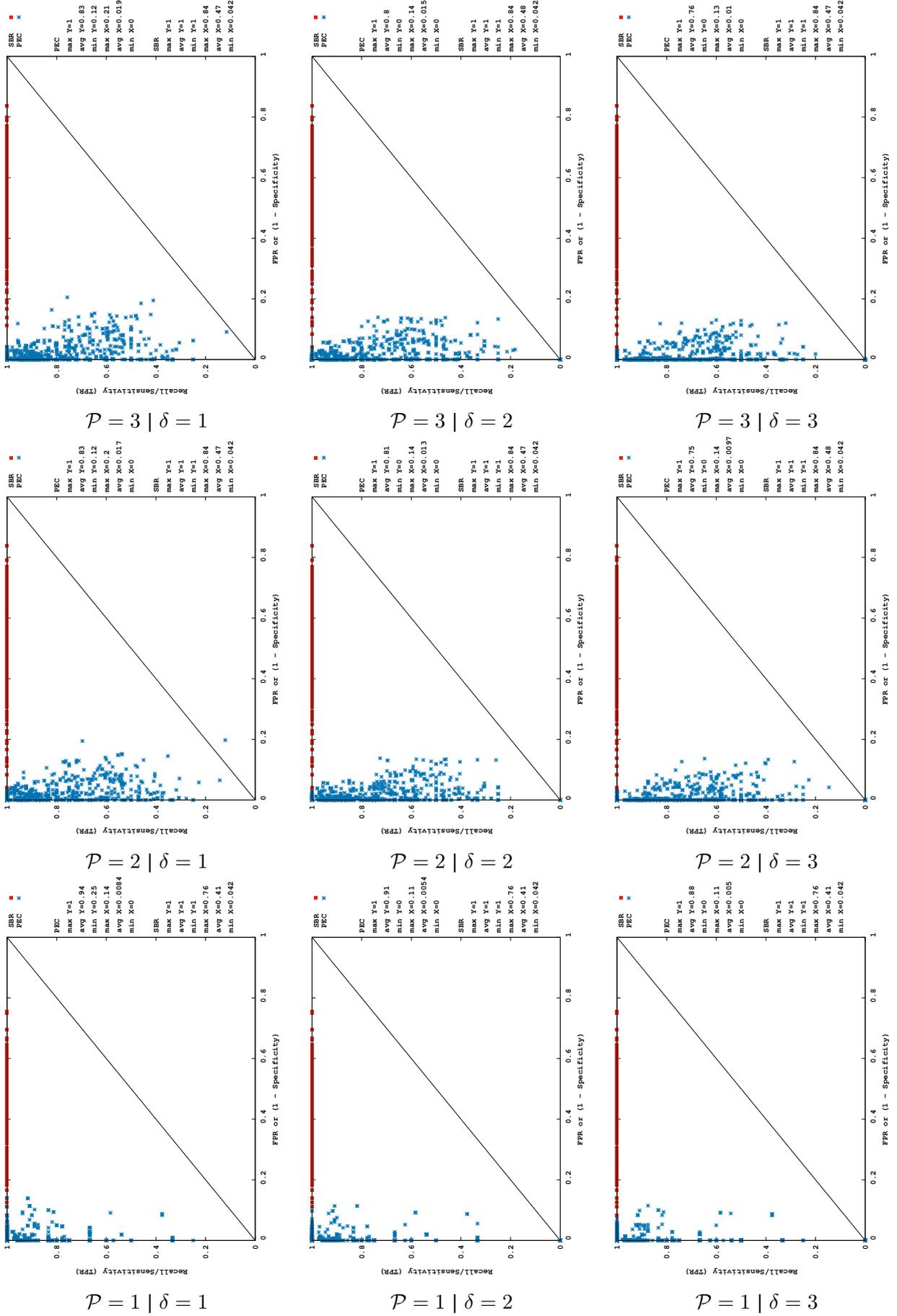


Figure A.8: ROC space graphs switching parallel planes (\mathcal{P}) and path length (δ) for subject one using partial dataset where all plans have defined hour.

Table A.5: PEC results for subject one using complete dataset where most plans have undefined hour

\mathcal{P}	δ	PPV	TPR	F ₁ score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.617	0.732	0.646	0.957	0.043	0.577	0.033	0.967	0.268	0.988	0.383	0.012	1360519
	2	0.730	0.698	0.701	0.967	0.033	0.600	0.020	0.980	0.302	0.986	0.270	0.014	1365184
	3	0.867	0.663	0.762	0.977	0.023	0.624	0.008	0.992	0.337	0.985	0.133	0.015	1372886
2	1	0.576	0.721	0.623	0.921	0.079	0.559	0.063	0.937	0.279	0.978	0.424	0.022	1728866
	2	0.678	0.690	0.663	0.939	0.061	0.599	0.041	0.959	0.310	0.975	0.322	0.025	1748419
	3	0.816	0.672	0.719	0.957	0.043	0.656	0.019	0.981	0.328	0.974	0.184	0.026	1701166
3	1	0.557	0.694	0.605	0.888	0.112	0.530	0.091	0.909	0.306	0.964	0.443	0.036	2028554
	2	0.661	0.664	0.648	0.912	0.088	0.575	0.060	0.940	0.336	0.960	0.339	0.040	1996923
	3	0.770	0.621	0.671	0.930	0.070	0.610	0.034	0.966	0.379	0.956	0.230	0.044	1994874

Table A.6: SBR results for subject one using complete dataset where most plans have undefined hour

\mathcal{P}	δ	PPV	TPR	F ₁ score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.232	1.000	0.355	0.742	0.258	0.394	0.270	0.730	0.000	1.000	0.768	0.000	114232
	2	0.235	1.000	0.358	0.740	0.260	0.394	0.273	0.727	0.000	1.000	0.765	0.000	113853
	3	0.237	1.000	0.361	0.743	0.257	0.397	0.269	0.731	0.000	1.000	0.763	0.000	116951
2	1	0.233	1.000	0.372	0.684	0.316	0.383	0.343	0.657	0.000	1.000	0.767	0.000	222851
	2	0.236	1.000	0.375	0.686	0.314	0.385	0.341	0.659	0.000	1.000	0.764	0.000	228615
	3	0.240	1.000	0.379	0.687	0.313	0.388	0.340	0.660	0.000	1.000	0.760	0.000	223573
3	1	0.261	1.000	0.410	0.649	0.351	0.391	0.396	0.604	0.000	1.000	0.739	0.000	349004
	2	0.264	1.000	0.414	0.654	0.346	0.395	0.391	0.609	0.000	1.000	0.736	0.000	345103
	3	0.261	1.000	0.410	0.650	0.350	0.391	0.395	0.605	0.000	1.000	0.739	0.000	342254

Table A.7: PEC results for subject one using partial dataset where most plans have undefined hour

\mathcal{P}	δ	PPV	TPR	F_1 score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.667	0.776	0.685	0.929	0.071	0.604	0.058	0.942	0.224	0.983	0.333	0.017	1218760
	2	0.757	0.776	0.743	0.947	0.053	0.657	0.039	0.961	0.224	0.983	0.243	0.017	1255829
	3	0.886	0.719	0.816	0.965	0.035	0.672	0.014	0.986	0.281	0.978	0.114	0.022	1234891
2	1	0.577	0.753	0.635	0.858	0.142	0.548	0.127	0.873	0.247	0.961	0.423	0.039	1550325
	2	0.693	0.719	0.687	0.893	0.107	0.602	0.081	0.919	0.281	0.957	0.307	0.043	1541239
	3	0.844	0.684	0.741	0.929	0.071	0.672	0.033	0.967	0.316	0.953	0.156	0.047	1555200
3	1	0.554	0.714	0.611	0.796	0.204	0.483	0.189	0.811	0.286	0.928	0.446	0.072	1784763
	2	0.659	0.683	0.657	0.838	0.162	0.543	0.129	0.871	0.317	0.926	0.341	0.074	1744339
	3	0.823	0.623	0.696	0.885	0.115	0.616	0.054	0.946	0.377	0.916	0.177	0.084	1741836

Table A.8: SBR results for subject one using partial dataset where most plans have undefined hour

\mathcal{P}	δ	PPV	TPR	F_1 score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.229	1.000	0.363	0.623	0.377	0.362	0.408	0.592	0.000	1.000	0.771	0.000	122224
	2	0.227	1.000	0.361	0.621	0.379	0.359	0.410	0.590	0.000	1.000	0.773	0.000	109288
	3	0.228	1.000	0.362	0.620	0.380	0.360	0.412	0.588	0.000	1.000	0.772	0.000	120262
2	1	0.261	1.000	0.409	0.544	0.456	0.341	0.529	0.471	0.000	1.000	0.739	0.000	241589
	2	0.262	1.000	0.411	0.549	0.451	0.345	0.523	0.477	0.000	1.000	0.738	0.000	229600
	3	0.262	1.000	0.411	0.552	0.448	0.346	0.520	0.480	0.000	1.000	0.738	0.000	212810
3	1	0.310	1.000	0.471	0.517	0.483	0.331	0.608	0.392	0.000	1.000	0.690	0.000	367108
	2	0.311	1.000	0.471	0.521	0.479	0.334	0.602	0.398	0.000	1.000	0.689	0.000	337625
	3	0.315	1.000	0.476	0.526	0.474	0.339	0.596	0.404	0.000	1.000	0.685	0.000	357385

Table A.9: PEC results for subject one using complete dataset where all plans have a defined hour

\mathcal{P}	δ	PPV	TPR	F_1 score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.949	0.941	0.937	0.993	0.007	0.915	0.005	0.995	0.059	0.997	0.051	0.003	1436552
	2	0.959	0.905	0.939	0.992	0.008	0.884	0.004	0.996	0.095	0.996	0.041	0.004	1228736
	3	0.971	0.879	0.928	0.992	0.008	0.864	0.002	0.998	0.121	0.995	0.029	0.005	1231246
2	1	0.911	0.778	0.821	0.974	0.026	0.777	0.010	0.990	0.222	0.982	0.089	0.018	1572000
	2	0.913	0.758	0.812	0.973	0.027	0.769	0.009	0.991	0.242	0.980	0.087	0.020	1591897
	3	0.944	0.704	0.781	0.972	0.028	0.736	0.006	0.994	0.296	0.977	0.056	0.023	1538830
3	1	0.878	0.748	0.788	0.966	0.034	0.737	0.015	0.985	0.252	0.978	0.122	0.022	1619256
	2	0.902	0.727	0.785	0.967	0.033	0.733	0.012	0.988	0.273	0.977	0.098	0.023	1642936
	3	0.922	0.694	0.765	0.967	0.033	0.719	0.008	0.992	0.306	0.974	0.078	0.026	1725874

Table A.10: SBR results for subject one using complete dataset where all plans have a defined hour

\mathcal{P}	δ	PPV	TPR	F_1 score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.247	1.000	0.373	0.754	0.246	0.408	0.258	0.742	0.000	1.000	0.753	0.000	140278
	2	0.233	1.000	0.358	0.744	0.256	0.395	0.268	0.732	0.000	1.000	0.767	0.000	116238
	3	0.237	1.000	0.362	0.744	0.256	0.398	0.268	0.732	0.000	1.000	0.763	0.000	106656
2	1	0.222	1.000	0.358	0.689	0.311	0.375	0.337	0.663	0.000	1.000	0.778	0.000	205344
	2	0.217	1.000	0.352	0.681	0.319	0.369	0.345	0.655	0.000	1.000	0.783	0.000	204669
	3	0.220	1.000	0.357	0.686	0.314	0.374	0.340	0.660	0.000	1.000	0.780	0.000	198778
3	1	0.225	1.000	0.362	0.676	0.324	0.373	0.353	0.647	0.000	1.000	0.775	0.000	223214
	2	0.227	1.000	0.365	0.678	0.322	0.375	0.352	0.648	0.000	1.000	0.773	0.000	245694
	3	0.221	1.000	0.357	0.672	0.328	0.368	0.358	0.642	0.000	1.000	0.779	0.000	256180

Table A.11: PEC results for subject one using partial dataset where all plans have a defined hour

\mathcal{P}	δ	PPV	TPR	F_1 score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.947	0.936	0.933	0.987	0.013	0.906	0.008	0.992	0.064	0.995	0.053	0.005	1086986
	2	0.962	0.912	0.939	0.988	0.012	0.892	0.005	0.995	0.088	0.993	0.038	0.007	1070131
	3	0.964	0.877	0.912	0.986	0.014	0.858	0.005	0.995	0.123	0.991	0.036	0.009	1173837
2	1	0.913	0.835	0.862	0.964	0.036	0.818	0.017	0.983	0.165	0.977	0.087	0.023	1278742
	2	0.929	0.812	0.860	0.965	0.035	0.808	0.013	0.987	0.188	0.975	0.071	0.025	1288031
	3	0.943	0.750	0.828	0.963	0.037	0.757	0.010	0.990	0.250	0.970	0.057	0.030	1336397
3	1	0.906	0.833	0.857	0.963	0.037	0.809	0.019	0.981	0.167	0.978	0.094	0.022	1276504
	2	0.918	0.803	0.854	0.962	0.038	0.794	0.015	0.985	0.197	0.974	0.082	0.026	1316860
	3	0.946	0.764	0.841	0.964	0.036	0.771	0.010	0.990	0.236	0.971	0.054	0.029	1310149

Table A.12: SBR results for subject one using partial dataset where all plans have a defined hour

\mathcal{P}	δ	PPV	TPR	F_1 score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.233	1.000	0.367	0.621	0.379	0.363	0.410	0.590	0.000	1.000	0.767	0.000	120488
	2	0.233	1.000	0.367	0.622	0.378	0.364	0.410	0.590	0.000	1.000	0.767	0.000	113738
	3	0.233	1.000	0.366	0.622	0.378	0.363	0.410	0.590	0.000	1.000	0.767	0.000	119907
2	1	0.247	1.000	0.388	0.582	0.418	0.352	0.471	0.529	0.000	1.000	0.753	0.000	181258
	2	0.247	1.000	0.389	0.579	0.421	0.351	0.475	0.525	0.000	1.000	0.753	0.000	205456
	3	0.244	1.000	0.383	0.573	0.427	0.345	0.481	0.519	0.000	1.000	0.756	0.000	200268
3	1	0.246	1.000	0.388	0.580	0.420	0.351	0.473	0.527	0.000	1.000	0.754	0.000	192237
	2	0.249	1.000	0.390	0.576	0.424	0.351	0.480	0.520	0.000	1.000	0.751	0.000	204340
	3	0.248	1.000	0.391	0.583	0.417	0.353	0.470	0.530	0.000	1.000	0.752	0.000	194750

APPENDIX B – SUBJECT TWO – TABLES AND GRAPHS

Table B.1: PEC, subject two, partial dataset, most plans with undefined hour

\mathcal{P}	δ	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.624	0.251	0.779	0.172	0.647	0.167	0.872	0.083	0.117	0.097	1247948
	2	0.757	0.238	0.772	0.186	0.729	0.155	0.912	0.063	0.071	0.075	1289799
	3	0.869	0.163	0.657	0.278	0.794	0.141	0.941	0.040	0.024	0.032	1368993
2	1	0.518	0.166	0.709	0.149	0.574	0.124	0.748	0.097	0.246	0.123	1528052
	2	0.624	0.186	0.679	0.142	0.621	0.120	0.791	0.088	0.185	0.114	1557904
	3	0.773	0.177	0.570	0.162	0.641	0.125	0.845	0.057	0.084	0.073	1548836
3	1	0.533	0.118	0.710	0.123	0.595	0.098	0.677	0.092	0.351	0.134	1590231
	2	0.602	0.134	0.663	0.116	0.616	0.098	0.710	0.086	0.283	0.124	1600159
	3	0.756	0.148	0.538	0.126	0.610	0.103	0.770	0.065	0.135	0.092	1632090

Table B.2: SBR, subject two, partial dataset, most plans with undefined hour

\mathcal{P}	δ	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.258	0.116	1.000	0.000	0.388	0.131	0.462	0.169	0.605	0.190	229622
	2	0.263	0.117	1.000	0.000	0.393	0.131	0.468	0.164	0.599	0.184	635004
	3	0.252	0.110	1.000	0.000	0.381	0.126	0.454	0.165	0.614	0.186	557124
2	1	0.287	0.054	1.000	0.000	0.440	0.062	0.406	0.096	0.754	0.125	564936
	2	0.287	0.055	1.000	0.000	0.441	0.062	0.403	0.095	0.758	0.123	530770
	3	0.287	0.055	1.000	0.000	0.441	0.062	0.402	0.096	0.760	0.124	580439
3	1	0.368	0.035	1.000	0.000	0.536	0.036	0.442	0.058	0.816	0.090	881584
	2	0.368	0.034	1.000	0.000	0.536	0.035	0.440	0.055	0.821	0.086	858697
	3	0.368	0.036	1.000	0.000	0.537	0.037	0.442	0.060	0.818	0.092	910819

Table B.3: PEC, subject two, partial dataset, all plans with defined hour

\mathcal{P}	δ	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.957	0.072	0.886	0.138	0.905	0.107	0.979	0.024	0.009	0.016	1197948
	2	0.970	0.053	0.866	0.161	0.912	0.102	0.980	0.023	0.006	0.011	1188977
	3	0.976	0.042	0.745	0.248	0.863	0.130	0.967	0.030	0.005	0.009	1229679
2	1	0.789	0.184	0.751	0.175	0.744	0.140	0.908	0.056	0.062	0.054	1416776
	2	0.883	0.143	0.724	0.197	0.772	0.139	0.927	0.048	0.031	0.038	1388242
	3	0.917	0.107	0.639	0.233	0.780	0.135	0.927	0.044	0.019	0.025	1449717
3	1	0.754	0.202	0.750	0.177	0.722	0.141	0.902	0.056	0.070	0.060	1389968
	2	0.843	0.157	0.734	0.183	0.760	0.124	0.921	0.045	0.042	0.044	1465347
	3	0.928	0.098	0.635	0.237	0.779	0.138	0.928	0.045	0.017	0.024	1370352

Table B.4: SBR, subject two, partial dataset, all plans with defined hour

\mathcal{P}	δ	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
		Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
1	1	0.250	0.111	1.000	0.000	0.378	0.126	0.453	0.167	0.615	0.188	240757
	2	0.247	0.109	1.000	0.000	0.376	0.126	0.445	0.161	0.625	0.181	244304
	3	0.257	0.119	1.000	0.000	0.386	0.135	0.451	0.165	0.618	0.185	251132
2	1	0.270	0.089	1.000	0.000	0.410	0.103	0.424	0.127	0.689	0.160	423821
	2	0.271	0.095	1.000	0.000	0.409	0.109	0.425	0.133	0.684	0.167	395310
	3	0.274	0.095	1.000	0.000	0.413	0.107	0.428	0.132	0.684	0.167	434274
3	1	0.270	0.097	1.000	0.000	0.407	0.111	0.426	0.137	0.682	0.170	425705
	2	0.272	0.095	1.000	0.000	0.410	0.108	0.429	0.133	0.679	0.165	444754
	3	0.275	0.090	1.000	0.000	0.415	0.103	0.430	0.125	0.680	0.159	427859

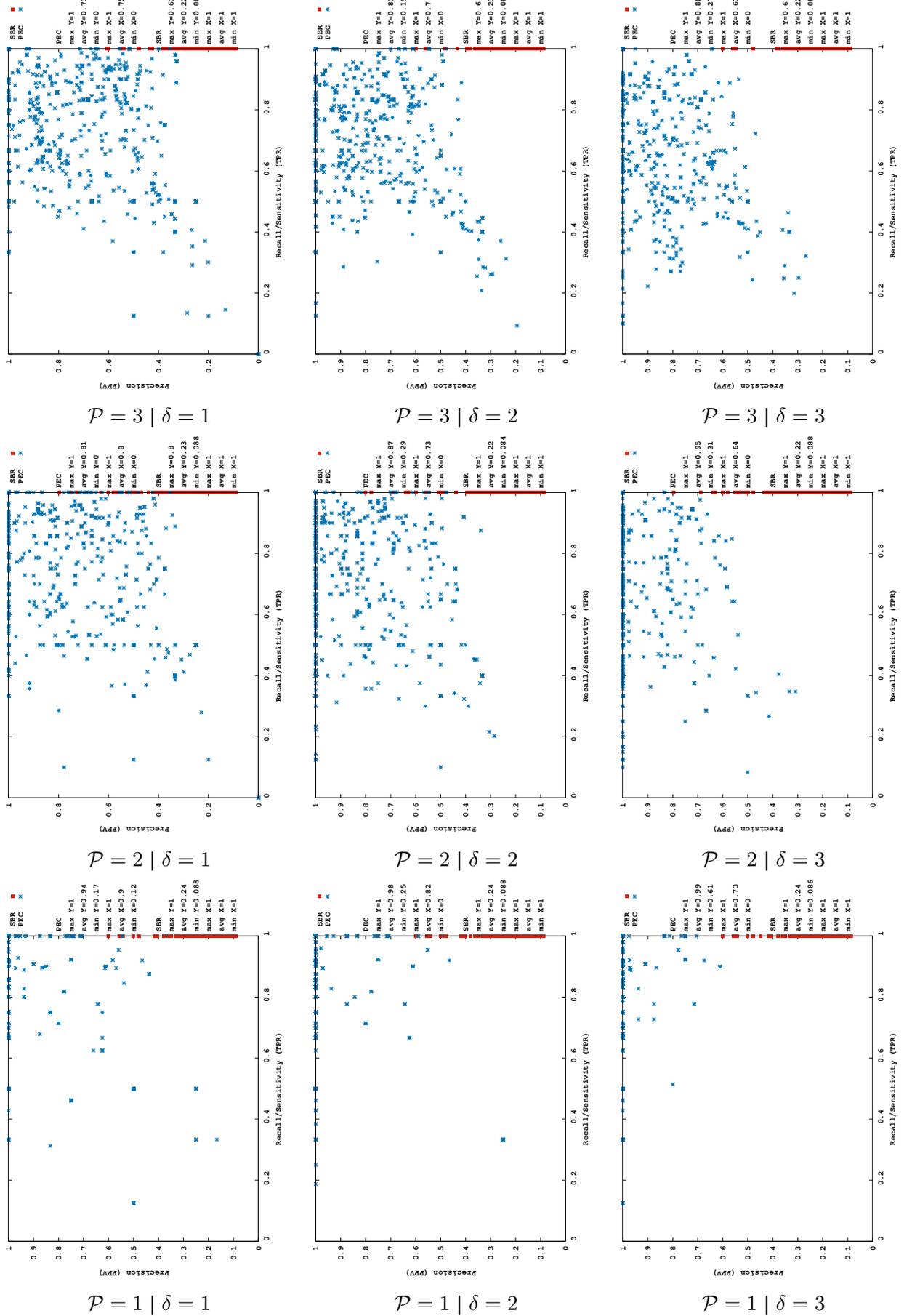


Figure B.2: PR space graphs switching parallel planes (\mathcal{P}) and path length (δ) for subject two using complete dataset where all plans have defined hour.

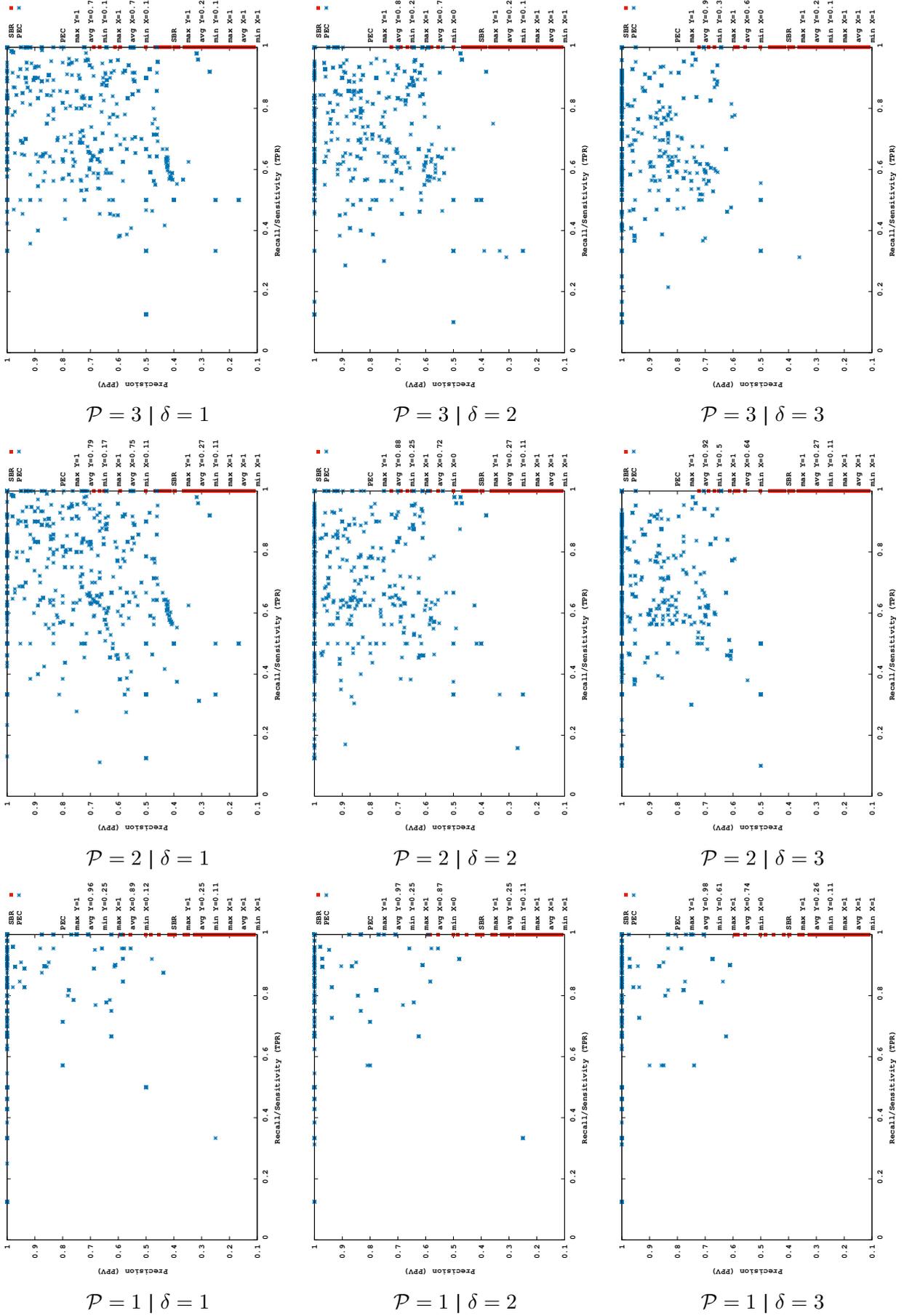


Figure B.4: PR space graphs switching parallel planes (\mathcal{P}) and path length (δ) for subject two using partial dataset where all plans have defined hour.

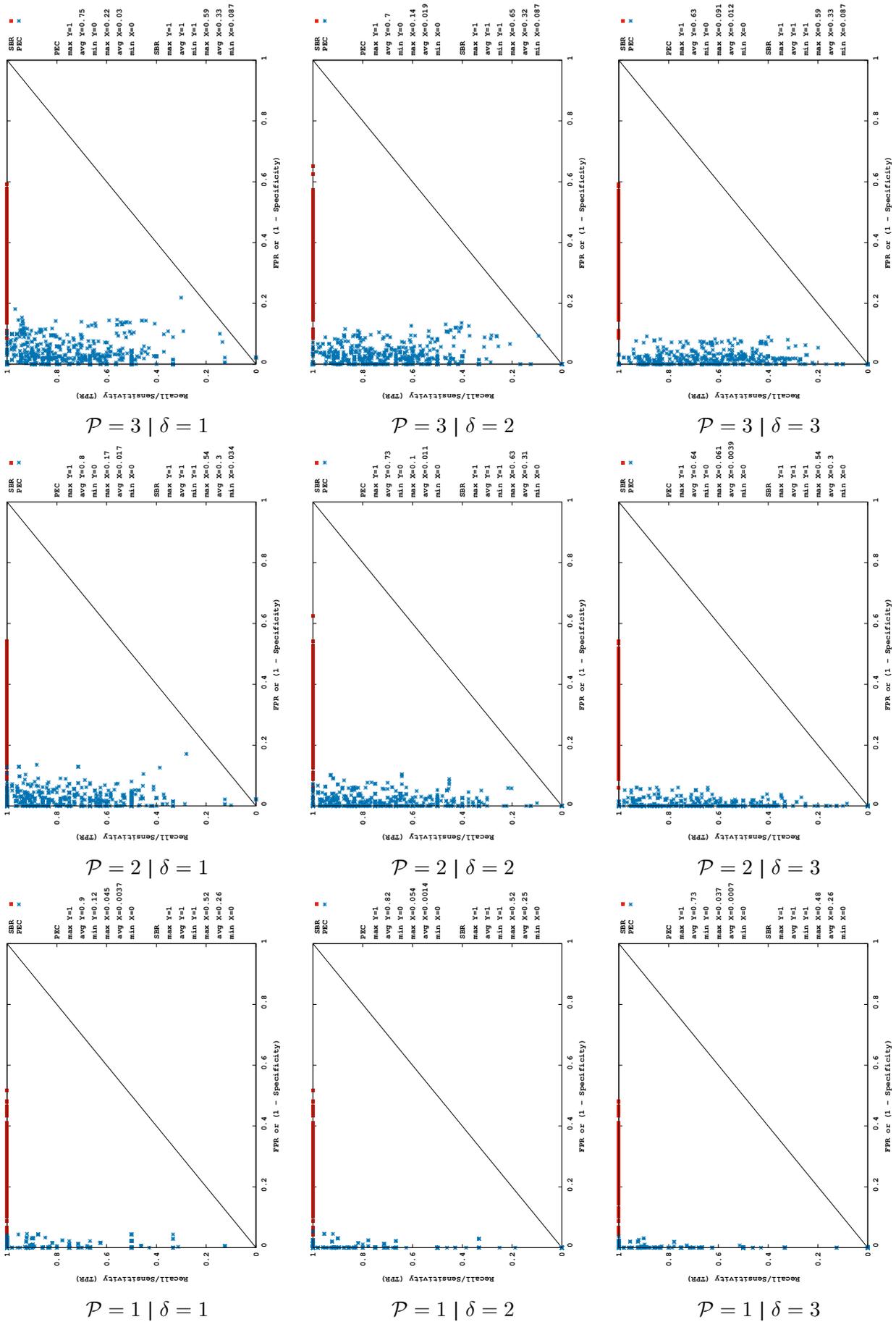


Figure B.6: ROC space graphs switching parallel planes (\mathcal{P}) and path length (δ) for subject two using complete dataset where all plans have defined hour.

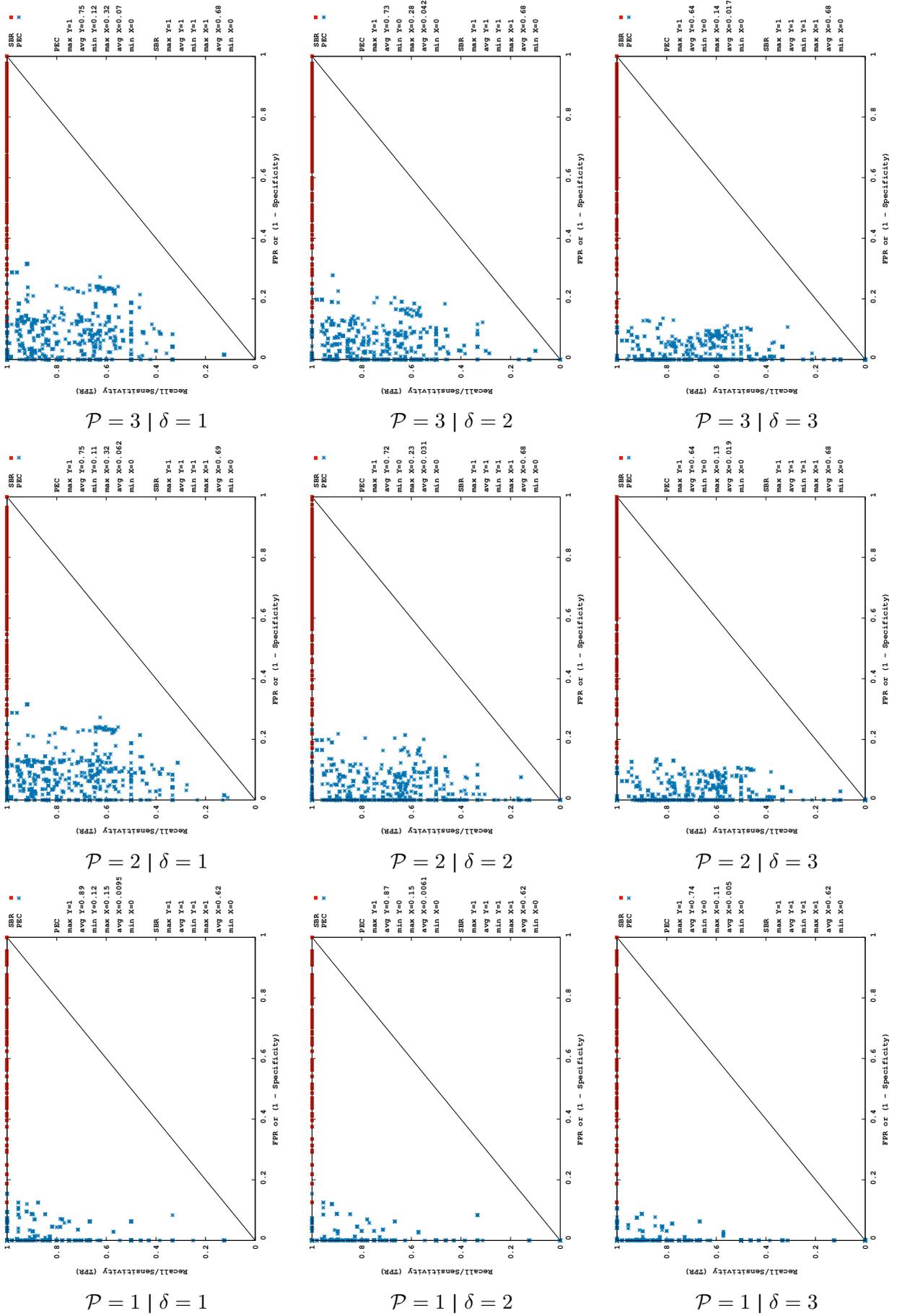


Figure B.8: ROC space graphs switching parallel planes (\mathcal{P}) and path length (δ) for subject two using partial dataset where all plans have defined hour.

Table B.5: PEC results for subject two using complete dataset where most plans have undefined hour

\mathcal{P}	δ	PPV	TPR	F ₁ score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.661	0.779	0.687	0.960	0.040	0.628	0.032	0.968	0.221	0.991	0.339	0.009	1600281
	2	0.785	0.689	0.731	0.970	0.030	0.599	0.018	0.982	0.311	0.987	0.215	0.013	1585851
	3	0.871	0.597	0.810	0.977	0.023	0.557	0.007	0.993	0.403	0.983	0.129	0.017	1554901
2	1	0.505	0.743	0.578	0.901	0.099	0.521	0.088	0.912	0.257	0.981	0.495	0.019	1975257
	2	0.686	0.697	0.663	0.937	0.063	0.596	0.045	0.955	0.303	0.977	0.314	0.023	1935193
	3	0.792	0.595	0.666	0.951	0.049	0.574	0.022	0.978	0.405	0.970	0.208	0.030	2039119
3	1	0.462	0.739	0.552	0.852	0.148	0.481	0.140	0.860	0.261	0.972	0.538	0.028	2317015
	2	0.594	0.696	0.619	0.894	0.106	0.543	0.087	0.913	0.304	0.967	0.406	0.033	2315691
	3	0.750	0.548	0.608	0.920	0.080	0.536	0.039	0.961	0.452	0.951	0.250	0.049	2293678

Table B.6: SBR results for subject two using complete dataset where most plans have undefined hour

\mathcal{P}	δ	PPV	TPR	F ₁ score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.248	1.000	0.370	0.755	0.245	0.406	0.256	0.744	0.000	1.000	0.752	0.000	203249
	2	0.239	1.000	0.361	0.753	0.247	0.399	0.258	0.742	0.000	1.000	0.761	0.000	222329
	3	0.237	1.000	0.359	0.749	0.251	0.396	0.262	0.738	0.000	1.000	0.763	0.000	726215
2	1	0.227	1.000	0.360	0.679	0.321	0.376	0.346	0.654	0.000	1.000	0.773	0.000	479012
	2	0.237	1.000	0.371	0.685	0.315	0.385	0.340	0.660	0.000	1.000	0.763	0.000	480977
	3	0.227	1.000	0.360	0.675	0.325	0.374	0.350	0.650	0.000	1.000	0.773	0.000	519392
3	1	0.245	1.000	0.389	0.637	0.363	0.376	0.407	0.593	0.000	1.000	0.755	0.000	782808
	2	0.246	1.000	0.390	0.635	0.365	0.376	0.409	0.591	0.000	1.000	0.754	0.000	845159
	3	0.246	1.000	0.390	0.639	0.361	0.377	0.405	0.595	0.000	1.000	0.754	0.000	829582

Table B.7: PEC results for subject two using partial dataset where most plans have undefined hour

\mathcal{P}	δ	PPV	TPR	F ₁ score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.624	0.779	0.647	0.872	0.128	0.564	0.117	0.883	0.221	0.974	0.376	0.026	1247948
	2	0.757	0.772	0.729	0.912	0.088	0.635	0.071	0.929	0.228	0.974	0.243	0.026	1289799
	3	0.869	0.657	0.794	0.941	0.059	0.603	0.024	0.976	0.343	0.962	0.131	0.038	1368993
2	1	0.518	0.709	0.574	0.748	0.252	0.426	0.246	0.754	0.291	0.921	0.482	0.079	1528052
	2	0.624	0.679	0.621	0.791	0.209	0.479	0.185	0.815	0.321	0.918	0.376	0.082	1557904
	3	0.773	0.570	0.641	0.845	0.155	0.513	0.084	0.916	0.430	0.898	0.227	0.102	1548836
3	1	0.533	0.710	0.595	0.677	0.323	0.360	0.351	0.649	0.290	0.861	0.467	0.139	1590231
	2	0.602	0.663	0.616	0.710	0.290	0.397	0.283	0.717	0.337	0.855	0.398	0.145	1600159
	3	0.756	0.538	0.610	0.770	0.230	0.451	0.135	0.865	0.462	0.824	0.244	0.176	1632090

Table B.8: SBR results for subject two using partial dataset where most plans have undefined hour

\mathcal{P}	δ	PPV	TPR	F ₁ score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.258	1.000	0.388	0.462	0.538	0.297	0.605	0.395	0.000	1.000	0.742	0.000	229622
	2	0.263	1.000	0.393	0.468	0.532	0.303	0.599	0.401	0.000	1.000	0.737	0.000	635004
	3	0.252	1.000	0.381	0.454	0.546	0.289	0.614	0.386	0.000	1.000	0.748	0.000	557124
2	1	0.287	1.000	0.440	0.406	0.594	0.215	0.754	0.246	0.000	1.000	0.713	0.000	564936
	2	0.287	1.000	0.441	0.403	0.597	0.211	0.758	0.242	0.000	1.000	0.713	0.000	530770
	3	0.287	1.000	0.441	0.402	0.598	0.210	0.760	0.240	0.000	1.000	0.713	0.000	580439
3	1	0.368	1.000	0.536	0.442	0.558	0.180	0.816	0.184	0.000	1.000	0.632	0.000	881584
	2	0.368	1.000	0.536	0.440	0.560	0.175	0.821	0.179	0.000	1.000	0.632	0.000	858697
	3	0.368	1.000	0.537	0.442	0.558	0.177	0.818	0.182	0.000	1.000	0.632	0.000	910819

Table B.9: PEC results for subject two using complete dataset where all plans have a defined hour

\mathcal{P}	δ	PPV	TPR	F_1 score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.942	0.901	0.912	0.992	0.008	0.882	0.004	0.996	0.099	0.996	0.058	0.004	1414605
	2	0.980	0.822	0.924	0.991	0.009	0.813	0.001	0.999	0.178	0.993	0.020	0.007	1429505
	3	0.990	0.733	0.899	0.988	0.012	0.728	0.001	0.999	0.267	0.989	0.010	0.011	1470727
2	1	0.813	0.801	0.793	0.973	0.027	0.746	0.017	0.983	0.199	0.988	0.187	0.012	1646426
	2	0.873	0.732	0.783	0.974	0.026	0.706	0.011	0.989	0.268	0.984	0.127	0.016	1769721
	3	0.949	0.636	0.807	0.976	0.024	0.651	0.004	0.996	0.364	0.979	0.051	0.021	1666964
3	1	0.733	0.745	0.719	0.956	0.044	0.660	0.030	0.970	0.255	0.983	0.267	0.017	1820602
	2	0.835	0.701	0.755	0.963	0.037	0.663	0.019	0.981	0.299	0.979	0.165	0.021	1919968
	3	0.877	0.630	0.764	0.965	0.035	0.624	0.012	0.988	0.370	0.974	0.123	0.026	1853750

Table B.10: SBR results for subject two using complete dataset where all plans have a defined hour

\mathcal{P}	δ	PPV	TPR	F_1 score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.244	1.000	0.366	0.755	0.245	0.403	0.256	0.744	0.000	1.000	0.756	0.000	412081
	2	0.245	1.000	0.367	0.756	0.244	0.404	0.255	0.745	0.000	1.000	0.755	0.000	185169
	3	0.239	1.000	0.361	0.752	0.248	0.398	0.259	0.741	0.000	1.000	0.761	0.000	217185
2	1	0.231	1.000	0.359	0.717	0.283	0.385	0.300	0.700	0.000	1.000	0.769	0.000	354345
	2	0.222	1.000	0.349	0.705	0.295	0.375	0.314	0.686	0.000	1.000	0.778	0.000	393015
	3	0.224	1.000	0.352	0.718	0.282	0.381	0.300	0.700	0.000	1.000	0.776	0.000	365251
3	1	0.225	1.000	0.355	0.695	0.305	0.373	0.329	0.671	0.000	1.000	0.775	0.000	470355
	2	0.233	1.000	0.366	0.705	0.295	0.383	0.318	0.682	0.000	1.000	0.767	0.000	518653
	3	0.224	1.000	0.355	0.696	0.304	0.373	0.328	0.672	0.000	1.000	0.776	0.000	511486

Table B.11: PEC results for subject two using partial dataset where all plans have a defined hour

\mathcal{P}	δ	PPV	TPR	F_1 score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.957	0.886	0.905	0.979	0.021	0.861	0.009	0.991	0.114	0.987	0.043	0.013	1197948
	2	0.970	0.866	0.912	0.980	0.020	0.851	0.006	0.994	0.134	0.985	0.030	0.015	1188977
	3	0.976	0.745	0.863	0.967	0.033	0.732	0.005	0.995	0.255	0.972	0.024	0.028	1229679
2	1	0.789	0.751	0.744	0.908	0.092	0.660	0.062	0.938	0.249	0.954	0.211	0.046	1416776
	2	0.883	0.724	0.772	0.927	0.073	0.692	0.031	0.969	0.276	0.949	0.117	0.051	1388242
	3	0.917	0.639	0.780	0.927	0.073	0.636	0.019	0.981	0.361	0.938	0.083	0.062	1449717
3	1	0.754	0.750	0.722	0.902	0.098	0.638	0.070	0.930	0.250	0.956	0.246	0.044	1389968
	2	0.843	0.734	0.760	0.921	0.079	0.677	0.042	0.958	0.266	0.953	0.157	0.047	1465347
	3	0.928	0.635	0.779	0.928	0.072	0.636	0.017	0.983	0.365	0.938	0.072	0.062	1370352

Table B.12: SBR results for subject two using partial dataset where all plans have a defined hour

\mathcal{P}	δ	PPV	TPR	F_1 score	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
1	1	0.250	1.000	0.378	0.453	0.547	0.288	0.615	0.385	0.000	1.000	0.750	0.000	240757
	2	0.247	1.000	0.376	0.445	0.555	0.282	0.625	0.375	0.000	1.000	0.753	0.000	244304
	3	0.257	1.000	0.386	0.451	0.549	0.290	0.618	0.382	0.000	1.000	0.743	0.000	251132
2	1	0.270	1.000	0.410	0.424	0.576	0.253	0.689	0.311	0.000	1.000	0.730	0.000	423821
	2	0.271	1.000	0.409	0.425	0.575	0.256	0.684	0.316	0.000	1.000	0.729	0.000	395310
	3	0.274	1.000	0.413	0.428	0.572	0.257	0.684	0.316	0.000	1.000	0.726	0.000	434274
3	1	0.270	1.000	0.407	0.426	0.574	0.256	0.682	0.318	0.000	1.000	0.730	0.000	425705
	2	0.272	1.000	0.410	0.429	0.571	0.259	0.679	0.321	0.000	1.000	0.728	0.000	444754
	3	0.275	1.000	0.415	0.430	0.570	0.260	0.680	0.320	0.000	1.000	0.725	0.000	427859

APPENDIX C – GENERATED PLAN LIBRARIES – TABLES AND GRAPHS

Table C.1: PEC results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$d$	$\$F$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	5	10	0.988	0.021	0.885	0.124	0.929	0.071	0.995	0.005	0.001	0.002	605000
		30	0.994	0.012	0.995	0.010	0.994	0.012	0.999	0.001	0.001	0.001	272500
		50	0.989	0.021	0.993	0.015	0.990	0.018	0.999	0.002	0.001	0.002	285000
	8	10	0.968	0.044	0.970	0.047	0.968	0.038	0.996	0.004	0.002	0.003	474286
		30	0.979	0.032	0.984	0.028	0.981	0.027	0.998	0.003	0.002	0.003	311429
		50	0.997	0.005	0.994	0.011	0.995	0.008	1.000	0.001	0.000	0.000	234286
	11	10	0.992	0.015	0.978	0.034	0.984	0.023	0.998	0.002	0.001	0.001	482000
		30	0.977	0.028	0.997	0.006	0.987	0.016	0.998	0.002	0.002	0.002	222000
		50	0.995	0.009	0.999	0.002	0.997	0.006	1.000	0.001	0.000	0.001	217000
50	5	10	0.993	0.014	0.880	0.146	0.924	0.090	0.997	0.003	0.000	0.000	955000
		30	0.968	0.049	0.948	0.084	0.953	0.058	0.998	0.003	0.001	0.002	467500
		50	0.986	0.025	0.970	0.053	0.976	0.039	0.999	0.002	0.001	0.001	417500
	8	10	0.959	0.057	0.907	0.087	0.929	0.055	0.997	0.003	0.002	0.002	1025714
		30	0.963	0.047	0.977	0.038	0.969	0.035	0.998	0.002	0.001	0.002	425714
		50	0.974	0.037	0.984	0.028	0.978	0.029	0.999	0.002	0.001	0.001	315714
	11	10	0.976	0.033	0.939	0.055	0.956	0.032	0.998	0.001	0.001	0.001	896000
		30	0.978	0.029	0.987	0.023	0.982	0.021	0.999	0.001	0.001	0.001	426000
		50	0.970	0.034	0.998	0.004	0.983	0.018	0.999	0.001	0.001	0.001	327000
75	5	10	0.989	0.020	0.775	0.131	0.856	0.076	0.997	0.002	0.000	0.000	1362500
		30	0.965	0.056	0.945	0.086	0.951	0.063	0.998	0.002	0.001	0.001	597500
		50	0.992	0.015	0.960	0.067	0.973	0.042	0.999	0.001	0.000	0.000	395000
	8	10	0.983	0.029	0.890	0.097	0.930	0.057	0.998	0.001	0.000	0.001	1310000
		30	0.976	0.037	0.970	0.049	0.972	0.036	0.999	0.001	0.001	0.001	617143
		50	0.972	0.037	0.986	0.026	0.978	0.026	0.999	0.001	0.001	0.001	405714
	11	10	0.976	0.032	0.926	0.052	0.949	0.027	0.998	0.001	0.001	0.001	1369000
		30	0.983	0.024	0.971	0.042	0.976	0.026	0.999	0.001	0.000	0.001	589000
		50	0.975	0.032	0.985	0.025	0.979	0.021	0.999	0.001	0.001	0.001	418000
100	5	10	0.953	0.077	0.698	0.186	0.782	0.149	0.996	0.002	0.001	0.001	2022500
		30	0.995	0.010	0.948	0.085	0.966	0.052	0.999	0.001	0.000	0.000	712500
		50	0.989	0.021	0.965	0.060	0.974	0.041	0.999	0.001	0.000	0.000	527500
	8	10	0.971	0.042	0.867	0.069	0.913	0.047	0.998	0.001	0.000	0.001	1777143
		30	0.957	0.054	0.960	0.058	0.957	0.046	0.999	0.001	0.001	0.001	737143
		50	0.987	0.022	0.984	0.028	0.985	0.023	1.000	0.001	0.000	0.000	515714
	11	10	0.982	0.027	0.930	0.056	0.954	0.030	0.999	0.001	0.000	0.000	1834000
		30	0.996	0.007	0.975	0.038	0.985	0.022	1.000	0.000	0.000	0.000	733000
		50	0.992	0.014	0.983	0.028	0.987	0.019	1.000	0.000	0.000	0.000	471000

Table C.2: PEC results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$d$	$\$F$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	5	10	0.937	0.074	0.792	0.120	0.849	0.090	0.980	0.011	0.005	0.006	1107500
		30	0.993	0.012	0.960	0.047	0.975	0.027	0.996	0.004	0.001	0.002	650893
		50	0.971	0.042	0.931	0.078	0.945	0.048	0.992	0.007	0.003	0.005	547857
	8	10	0.960	0.041	0.927	0.058	0.941	0.044	0.991	0.007	0.004	0.004	975101
		30	0.971	0.034	0.959	0.060	0.960	0.044	0.994	0.006	0.003	0.003	557637
		50	0.996	0.008	0.942	0.039	0.967	0.023	0.995	0.003	0.000	0.001	647088
	11	10	0.985	0.023	0.955	0.050	0.968	0.032	0.995	0.005	0.002	0.003	847798
		30	0.980	0.027	0.958	0.044	0.965	0.032	0.995	0.005	0.002	0.003	598711
		50	0.997	0.005	0.975	0.039	0.983	0.027	0.998	0.003	0.000	0.000	438395
50	5	10	0.937	0.072	0.762	0.113	0.833	0.092	0.989	0.005	0.002	0.002	1854702
		30	0.941	0.064	0.848	0.103	0.886	0.075	0.992	0.005	0.003	0.003	986964
		50	0.987	0.022	0.908	0.066	0.944	0.041	0.996	0.003	0.001	0.001	799643
	8	10	0.904	0.068	0.798	0.093	0.843	0.072	0.989	0.005	0.004	0.003	1886868
		30	0.955	0.044	0.921	0.061	0.934	0.047	0.995	0.004	0.002	0.002	862179
		50	0.959	0.040	0.937	0.057	0.945	0.042	0.995	0.003	0.002	0.002	704176
	11	10	0.916	0.061	0.872	0.052	0.891	0.043	0.992	0.004	0.004	0.003	1704158
		30	0.968	0.032	0.943	0.040	0.953	0.031	0.996	0.002	0.002	0.002	854249
		50	0.960	0.036	0.975	0.025	0.967	0.023	0.997	0.002	0.002	0.002	710208
75	5	10	0.899	0.101	0.659	0.123	0.751	0.106	0.990	0.004	0.002	0.002	2570655
		30	0.950	0.061	0.827	0.096	0.878	0.068	0.994	0.003	0.001	0.002	1186429
		50	0.986	0.024	0.858	0.081	0.914	0.047	0.996	0.002	0.000	0.001	945179
	8	10	0.882	0.079	0.778	0.088	0.822	0.069	0.992	0.003	0.003	0.002	2478306
		30	0.955	0.048	0.892	0.064	0.919	0.050	0.996	0.002	0.001	0.002	1131832
		50	0.971	0.036	0.934	0.046	0.951	0.031	0.997	0.002	0.001	0.001	906978
	11	10	0.930	0.055	0.837	0.084	0.877	0.063	0.994	0.003	0.002	0.001	2330854
		30	0.977	0.027	0.921	0.036	0.947	0.024	0.997	0.001	0.001	0.001	1125268
		50	0.961	0.032	0.951	0.045	0.953	0.035	0.997	0.002	0.001	0.001	817830
100	5	10	0.845	0.135	0.551	0.127	0.657	0.125	0.991	0.003	0.002	0.002	3514107
		30	0.980	0.034	0.812	0.083	0.883	0.062	0.996	0.002	0.000	0.001	1445000
		50	0.980	0.031	0.864	0.075	0.916	0.046	0.997	0.002	0.000	0.001	1030179
	8	10	0.896	0.092	0.721	0.110	0.794	0.098	0.994	0.003	0.002	0.001	3302747
		30	0.949	0.046	0.888	0.059	0.914	0.050	0.997	0.002	0.001	0.001	1344725
		50	0.977	0.029	0.922	0.041	0.948	0.027	0.998	0.001	0.001	0.001	1043727
	11	10	0.938	0.045	0.818	0.078	0.869	0.054	0.995	0.002	0.001	0.001	3031313
		30	0.981	0.025	0.926	0.036	0.952	0.025	0.998	0.001	0.000	0.001	1315395
		50	0.987	0.019	0.929	0.041	0.954	0.029	0.998	0.001	0.000	0.000	933026

Table C.3: PEC results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$d$	$\$F$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	5	10	0.910	0.070	0.726	0.113	0.799	0.084	0.966	0.014	0.008	0.008	1323732
		30	0.997	0.005	0.944	0.056	0.968	0.031	0.994	0.005	0.000	0.001	698328
		50	0.987	0.023	0.921	0.069	0.949	0.046	0.990	0.009	0.002	0.003	757045
	8	10	0.960	0.042	0.874	0.074	0.912	0.054	0.983	0.010	0.005	0.005	1142979
		30	0.967	0.037	0.930	0.089	0.938	0.068	0.991	0.009	0.003	0.004	615487
		50	0.996	0.007	0.914	0.055	0.950	0.035	0.990	0.007	0.001	0.001	730897
	11	10	0.996	0.008	0.930	0.057	0.960	0.033	0.994	0.005	0.000	0.001	840305
		30	0.987	0.015	0.943	0.040	0.962	0.024	0.992	0.005	0.002	0.002	779858
		50	1.000	0.000	0.951	0.075	0.969	0.048	0.995	0.008	0.000	0.000	509839
50	5	10	0.899	0.090	0.682	0.124	0.769	0.105	0.981	0.008	0.004	0.003	2320626
		30	0.948	0.054	0.833	0.084	0.883	0.058	0.989	0.005	0.003	0.003	1233273
		50	0.985	0.024	0.833	0.096	0.897	0.067	0.990	0.006	0.001	0.002	1012788
	8	10	0.835	0.083	0.680	0.112	0.744	0.096	0.978	0.007	0.007	0.003	2415905
		30	0.960	0.037	0.903	0.067	0.927	0.044	0.992	0.004	0.003	0.003	1143031
		50	0.959	0.035	0.933	0.053	0.944	0.037	0.994	0.004	0.003	0.002	929375
	11	10	0.871	0.070	0.804	0.067	0.832	0.059	0.984	0.005	0.007	0.004	2246629
		30	0.963	0.034	0.925	0.045	0.942	0.035	0.994	0.004	0.003	0.002	1107689
		50	0.950	0.041	0.952	0.034	0.949	0.029	0.994	0.003	0.004	0.003	931262
75	5	10	0.839	0.123	0.548	0.105	0.652	0.098	0.982	0.005	0.004	0.003	3182197
		30	0.943	0.062	0.787	0.097	0.852	0.068	0.991	0.004	0.002	0.002	1520985
		50	0.982	0.030	0.771	0.102	0.857	0.069	0.992	0.004	0.001	0.001	1150227
	8	10	0.842	0.092	0.673	0.121	0.740	0.102	0.985	0.005	0.005	0.003	3212126
		30	0.951	0.044	0.858	0.071	0.899	0.053	0.993	0.004	0.002	0.002	1460459
		50	0.977	0.028	0.902	0.044	0.936	0.028	0.996	0.002	0.001	0.001	1114648
	11	10	0.874	0.086	0.754	0.088	0.807	0.080	0.989	0.004	0.004	0.002	3168019
		30	0.973	0.027	0.918	0.028	0.944	0.021	0.996	0.002	0.001	0.001	1441067
		50	0.970	0.024	0.953	0.030	0.961	0.021	0.997	0.002	0.001	0.001	1085123
100	5	10	0.807	0.144	0.452	0.107	0.567	0.102	0.984	0.003	0.003	0.002	4515025
		30	0.961	0.051	0.774	0.089	0.853	0.065	0.994	0.003	0.001	0.001	1823182
		50	0.978	0.033	0.806	0.094	0.877	0.065	0.994	0.003	0.001	0.001	1349470
	8	10	0.834	0.106	0.630	0.105	0.711	0.099	0.988	0.004	0.003	0.002	4387130
		30	0.949	0.032	0.867	0.065	0.903	0.048	0.995	0.002	0.002	0.001	1797639
		50	0.979	0.025	0.885	0.042	0.928	0.027	0.996	0.001	0.001	0.001	1335437
	11	10	0.897	0.062	0.754	0.081	0.815	0.068	0.992	0.003	0.002	0.001	4115618
		30	0.972	0.028	0.883	0.047	0.923	0.034	0.996	0.002	0.001	0.001	1761252
		50	0.985	0.019	0.899	0.062	0.935	0.042	0.997	0.002	0.001	0.001	1252194

Table C.4: SBR results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$d$	$\$F$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	5	10	0.456	0.093	1.000	0.000	0.618	0.086	0.928	0.018	0.075	0.019	40000
		30	0.774	0.115	1.000	0.000	0.866	0.073	0.977	0.015	0.024	0.015	17500
		50	0.857	0.078	1.000	0.000	0.920	0.047	0.988	0.007	0.013	0.007	47500
	8	10	0.498	0.077	1.000	0.000	0.658	0.069	0.934	0.018	0.068	0.019	38571
		30	0.730	0.076	1.000	0.000	0.841	0.052	0.974	0.007	0.027	0.008	31429
		50	0.832	0.077	1.000	0.000	0.905	0.047	0.986	0.007	0.015	0.007	50000
	11	10	0.497	0.078	1.000	0.000	0.658	0.070	0.935	0.013	0.068	0.014	77000
		30	0.762	0.067	1.000	0.000	0.862	0.043	0.978	0.007	0.023	0.007	43000
		50	0.851	0.059	1.000	0.000	0.918	0.035	0.987	0.006	0.013	0.006	45000
50	5	10	0.246	0.036	1.000	0.000	0.393	0.046	0.926	0.012	0.075	0.013	50000
		30	0.586	0.113	1.000	0.000	0.730	0.091	0.976	0.008	0.024	0.008	42500
		50	0.694	0.120	1.000	0.000	0.810	0.085	0.983	0.007	0.017	0.007	42500
	8	10	0.216	0.024	1.000	0.000	0.355	0.032	0.917	0.010	0.085	0.010	94286
		30	0.582	0.087	1.000	0.000	0.730	0.068	0.976	0.007	0.025	0.007	40000
		50	0.708	0.095	1.000	0.000	0.824	0.066	0.985	0.006	0.016	0.006	45714
	11	10	0.247	0.032	1.000	0.000	0.395	0.040	0.925	0.009	0.077	0.009	154000
		30	0.584	0.067	1.000	0.000	0.733	0.053	0.975	0.005	0.025	0.005	87000
		50	0.673	0.075	1.000	0.000	0.801	0.054	0.983	0.004	0.017	0.004	73000
75	5	10	0.171	0.029	1.000	0.000	0.291	0.042	0.923	0.011	0.078	0.011	80000
		30	0.419	0.089	1.000	0.000	0.583	0.088	0.974	0.007	0.027	0.008	62500
		50	0.614	0.127	1.000	0.000	0.750	0.098	0.986	0.006	0.015	0.006	47500
	8	10	0.169	0.016	1.000	0.000	0.288	0.023	0.925	0.008	0.076	0.008	102857
		30	0.434	0.076	1.000	0.000	0.599	0.073	0.973	0.006	0.027	0.006	50000
		50	0.578	0.097	1.000	0.000	0.725	0.078	0.984	0.004	0.016	0.005	52857
	11	10	0.164	0.016	1.000	0.000	0.281	0.023	0.925	0.007	0.076	0.007	221000
		30	0.475	0.071	1.000	0.000	0.639	0.066	0.975	0.005	0.025	0.005	99000
		50	0.592	0.068	1.000	0.000	0.740	0.054	0.985	0.003	0.015	0.003	105000
100	5	10	0.113	0.018	1.000	0.000	0.203	0.028	0.912	0.012	0.088	0.012	67500
		30	0.346	0.071	1.000	0.000	0.507	0.076	0.974	0.005	0.026	0.005	50000
		50	0.523	0.108	1.000	0.000	0.677	0.093	0.984	0.005	0.016	0.005	60000
	8	10	0.120	0.010	1.000	0.000	0.214	0.015	0.921	0.007	0.080	0.007	185714
		30	0.343	0.060	1.000	0.000	0.506	0.067	0.974	0.005	0.027	0.005	82857
		50	0.502	0.082	1.000	0.000	0.663	0.072	0.984	0.004	0.016	0.004	62857
	11	10	0.127	0.013	1.000	0.000	0.225	0.020	0.921	0.007	0.079	0.007	329000
		30	0.358	0.054	1.000	0.000	0.524	0.059	0.974	0.004	0.026	0.004	156000
		50	0.565	0.081	1.000	0.000	0.717	0.067	0.986	0.003	0.014	0.003	118000

Table C.5: SBR results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$d$	$\$F$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	5	10	0.451	0.069	1.000	0.000	0.616	0.064	0.892	0.023	0.117	0.025	42143
		30	0.783	0.101	1.000	0.000	0.873	0.065	0.968	0.017	0.034	0.019	33750
		50	0.845	0.078	1.000	0.000	0.913	0.047	0.980	0.011	0.021	0.011	51250
	8	10	0.451	0.063	1.000	0.000	0.617	0.060	0.888	0.022	0.120	0.023	50559
		30	0.712	0.080	1.000	0.000	0.827	0.055	0.964	0.013	0.039	0.015	47308
		50	0.783	0.061	1.000	0.000	0.876	0.039	0.972	0.009	0.030	0.010	45604
	11	10	0.455	0.076	1.000	0.000	0.620	0.071	0.903	0.031	0.103	0.034	92044
		30	0.696	0.057	1.000	0.000	0.819	0.040	0.956	0.011	0.047	0.012	56842
		50	0.834	0.053	1.000	0.000	0.908	0.032	0.981	0.007	0.021	0.008	45184
50	5	10	0.230	0.028	1.000	0.000	0.372	0.037	0.866	0.017	0.139	0.018	79524
		30	0.518	0.084	1.000	0.000	0.676	0.074	0.953	0.013	0.049	0.013	58036
		50	0.647	0.085	1.000	0.000	0.781	0.063	0.971	0.009	0.031	0.009	75536
	8	10	0.211	0.027	1.000	0.000	0.347	0.037	0.856	0.017	0.149	0.018	119560
		30	0.519	0.070	1.000	0.000	0.679	0.061	0.956	0.010	0.045	0.010	75229
		50	0.652	0.072	1.000	0.000	0.785	0.053	0.973	0.007	0.028	0.007	72756
	11	10	0.212	0.026	1.000	0.000	0.349	0.035	0.864	0.013	0.140	0.014	239079
		30	0.504	0.072	1.000	0.000	0.666	0.063	0.954	0.010	0.048	0.011	111719
		50	0.603	0.062	1.000	0.000	0.750	0.049	0.969	0.006	0.032	0.007	89661
75	5	10	0.160	0.022	1.000	0.000	0.274	0.032	0.869	0.015	0.134	0.016	125060
		30	0.403	0.063	1.000	0.000	0.570	0.064	0.957	0.009	0.044	0.009	58750
		50	0.556	0.069	1.000	0.000	0.711	0.056	0.975	0.006	0.026	0.006	71786
	8	10	0.152	0.019	1.000	0.000	0.264	0.028	0.863	0.012	0.140	0.012	192436
		30	0.370	0.044	1.000	0.000	0.537	0.047	0.951	0.007	0.050	0.008	94460
		50	0.514	0.067	1.000	0.000	0.675	0.059	0.970	0.006	0.030	0.006	81374
	11	10	0.149	0.019	1.000	0.000	0.258	0.029	0.862	0.010	0.141	0.010	374813
		30	0.395	0.046	1.000	0.000	0.564	0.047	0.952	0.006	0.049	0.007	163771
		50	0.543	0.066	1.000	0.000	0.701	0.056	0.973	0.005	0.028	0.006	121699
100	5	10	0.110	0.016	1.000	0.000	0.198	0.026	0.850	0.016	0.152	0.016	156964
		30	0.332	0.053	1.000	0.000	0.495	0.059	0.956	0.009	0.045	0.009	81250
		50	0.494	0.095	1.000	0.000	0.654	0.084	0.974	0.007	0.026	0.007	91786
	8	10	0.109	0.012	1.000	0.000	0.197	0.019	0.853	0.011	0.150	0.011	280604
		30	0.320	0.045	1.000	0.000	0.482	0.051	0.953	0.006	0.048	0.006	118462
		50	0.449	0.067	1.000	0.000	0.615	0.064	0.971	0.006	0.030	0.006	107967
	11	10	0.111	0.015	1.000	0.000	0.200	0.024	0.857	0.011	0.145	0.011	465595
		30	0.317	0.035	1.000	0.000	0.479	0.040	0.952	0.006	0.049	0.006	191395
		50	0.497	0.051	1.000	0.000	0.661	0.046	0.975	0.004	0.026	0.004	124874

Table C.6: SBR results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$d$	$\$F$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	5	10	0.432	0.066	1.000	0.000	0.599	0.064	0.855	0.029	0.161	0.033	65394
		30	0.785	0.089	1.000	0.000	0.876	0.055	0.963	0.019	0.041	0.021	46242
		50	0.858	0.073	1.000	0.000	0.921	0.043	0.977	0.013	0.026	0.014	58485
	8	10	0.439	0.054	1.000	0.000	0.607	0.052	0.855	0.026	0.160	0.029	74722
		30	0.712	0.076	1.000	0.000	0.828	0.052	0.957	0.020	0.047	0.023	48830
		50	0.744	0.047	1.000	0.000	0.852	0.031	0.959	0.010	0.046	0.011	51570
	11	10	0.455	0.080	1.000	0.000	0.619	0.074	0.897	0.038	0.111	0.042	106635
		30	0.683	0.056	1.000	0.000	0.810	0.040	0.942	0.013	0.065	0.014	77517
		50	0.805	0.054	1.000	0.000	0.890	0.034	0.974	0.010	0.028	0.011	64980
50	5	10	0.216	0.023	1.000	0.000	0.354	0.031	0.809	0.022	0.201	0.024	124354
		30	0.502	0.063	1.000	0.000	0.665	0.056	0.935	0.015	0.069	0.016	77197
		50	0.607	0.075	1.000	0.000	0.751	0.058	0.956	0.011	0.047	0.012	86727
	8	10	0.205	0.025	1.000	0.000	0.339	0.035	0.802	0.017	0.208	0.017	179900
		30	0.488	0.059	1.000	0.000	0.652	0.053	0.934	0.012	0.069	0.012	81495
		50	0.628	0.067	1.000	0.000	0.768	0.051	0.962	0.009	0.041	0.010	81041
	11	10	0.204	0.023	1.000	0.000	0.338	0.032	0.810	0.016	0.199	0.017	310522
		30	0.488	0.047	1.000	0.000	0.654	0.043	0.936	0.010	0.068	0.011	141931
		50	0.591	0.050	1.000	0.000	0.741	0.040	0.956	0.007	0.046	0.007	107617
75	5	10	0.163	0.022	1.000	0.000	0.280	0.032	0.821	0.022	0.185	0.023	152955
		30	0.380	0.055	1.000	0.000	0.547	0.057	0.938	0.012	0.065	0.012	75303
		50	0.528	0.067	1.000	0.000	0.688	0.057	0.963	0.008	0.039	0.008	86000
	8	10	0.149	0.016	1.000	0.000	0.260	0.024	0.807	0.014	0.199	0.014	232316
		30	0.357	0.043	1.000	0.000	0.524	0.046	0.931	0.009	0.072	0.010	134708
		50	0.503	0.061	1.000	0.000	0.666	0.054	0.959	0.008	0.043	0.008	100739
	11	10	0.138	0.016	1.000	0.000	0.243	0.025	0.804	0.015	0.202	0.016	475909
		30	0.379	0.036	1.000	0.000	0.548	0.038	0.934	0.008	0.068	0.009	207710
		50	0.515	0.047	1.000	0.000	0.678	0.041	0.961	0.006	0.040	0.006	150975
100	5	10	0.113	0.016	1.000	0.000	0.203	0.026	0.793	0.020	0.213	0.020	179056
		30	0.317	0.042	1.000	0.000	0.479	0.048	0.938	0.010	0.063	0.010	96995
		50	0.466	0.065	1.000	0.000	0.631	0.059	0.962	0.009	0.039	0.009	107076
	8	10	0.107	0.012	1.000	0.000	0.193	0.019	0.793	0.016	0.213	0.016	367814
		30	0.309	0.039	1.000	0.000	0.470	0.045	0.932	0.008	0.069	0.008	147138
		50	0.417	0.047	1.000	0.000	0.586	0.047	0.957	0.006	0.044	0.006	120311
	11	10	0.106	0.013	1.000	0.000	0.191	0.021	0.798	0.013	0.206	0.013	612724
		30	0.293	0.032	1.000	0.000	0.452	0.038	0.931	0.008	0.070	0.008	243112
		50	0.455	0.042	1.000	0.000	0.624	0.040	0.963	0.005	0.038	0.005	161793

Table C.7: PEC results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$f$	$\$c$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	1	1	0.995	0.008	0.969	0.048	0.981	0.028	0.998	0.002	0.000	0.001	1069000
		2	0.992	0.015	0.978	0.034	0.984	0.023	0.998	0.002	0.001	0.001	482000
		3	0.963	0.039	0.970	0.044	0.965	0.034	0.996	0.003	0.003	0.003	421000
	2	1	0.987	0.020	0.989	0.020	0.988	0.017	0.999	0.002	0.001	0.002	304000
		2	0.998	0.005	1.000	0.000	0.999	0.002	1.000	0.000	0.000	0.000	153000
		3	0.998	0.004	1.000	0.000	0.999	0.002	1.000	0.000	0.000	0.000	151000
	3	1	0.977	0.028	0.994	0.011	0.985	0.018	0.998	0.002	0.002	0.002	205000
		2	0.997	0.006	1.000	0.000	0.998	0.003	1.000	0.000	0.000	0.000	141000
		3	1.000	0.000	0.996	0.008	0.998	0.004	1.000	0.000	0.000	0.000	113000
50	1	1	0.945	0.049	0.870	0.073	0.903	0.045	0.995	0.002	0.002	0.002	2110000
		2	0.976	0.033	0.939	0.055	0.956	0.032	0.998	0.001	0.001	0.001	896000
		3	0.982	0.026	0.989	0.020	0.985	0.019	0.999	0.001	0.001	0.001	687000
	2	1	0.975	0.033	0.983	0.029	0.978	0.025	0.999	0.002	0.001	0.001	535000
		2	0.979	0.027	0.996	0.008	0.987	0.015	0.999	0.001	0.001	0.001	264000
		3	0.993	0.012	1.000	0.000	0.996	0.006	1.000	0.000	0.000	0.000	167000
	3	1	0.976	0.033	0.989	0.020	0.982	0.022	0.999	0.001	0.001	0.001	280000
		2	0.998	0.005	1.000	0.000	0.999	0.002	1.000	0.000	0.000	0.000	179000
		3	0.999	0.002	1.000	0.000	0.999	0.001	1.000	0.000	0.000	0.000	180000
75	1	1	0.943	0.060	0.858	0.067	0.896	0.052	0.997	0.002	0.001	0.001	2934000
		2	0.976	0.032	0.926	0.052	0.949	0.027	0.998	0.001	0.001	0.001	1369000
		3	0.994	0.010	0.980	0.032	0.987	0.019	1.000	0.001	0.000	0.000	947000
	2	1	0.981	0.027	0.964	0.048	0.971	0.028	0.999	0.001	0.000	0.001	638000
		2	0.990	0.017	0.990	0.018	0.990	0.016	1.000	0.001	0.000	0.000	303000
		3	0.992	0.013	0.996	0.008	0.994	0.010	1.000	0.000	0.000	0.000	203000
	3	1	0.986	0.022	0.991	0.016	0.988	0.017	1.000	0.001	0.000	0.001	352000
		2	0.996	0.008	0.998	0.004	0.997	0.006	1.000	0.000	0.000	0.000	170000
		3	0.999	0.002	1.000	0.000	0.999	0.001	1.000	0.000	0.000	0.000	163000
100	1	1	0.976	0.036	0.847	0.081	0.904	0.052	0.998	0.001	0.000	0.001	3591000
		2	0.982	0.027	0.930	0.056	0.954	0.030	0.999	0.001	0.000	0.000	1834000
		3	0.986	0.022	0.940	0.056	0.961	0.032	0.999	0.001	0.000	0.000	1292000
	2	1	0.979	0.029	0.966	0.050	0.971	0.031	0.999	0.001	0.000	0.001	915000
		2	0.969	0.033	0.978	0.035	0.973	0.025	0.999	0.001	0.001	0.001	361000
		3	0.983	0.023	0.998	0.004	0.990	0.013	1.000	0.000	0.000	0.000	232000
	3	1	0.979	0.031	0.972	0.040	0.975	0.028	0.999	0.001	0.000	0.001	464000
		2	0.993	0.013	0.999	0.002	0.996	0.007	1.000	0.000	0.000	0.000	180000
		3	0.999	0.003	0.999	0.002	0.999	0.002	1.000	0.000	0.000	0.000	178000

Table C.8: PEC results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$f$	$\$c$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	1	1	0.968	0.042	0.898	0.092	0.927	0.066	0.990	0.009	0.003	0.003	1630281
		2	0.985	0.023	0.955	0.050	0.968	0.032	0.995	0.005	0.002	0.003	847798
		3	0.931	0.054	0.929	0.056	0.928	0.048	0.988	0.008	0.007	0.005	795936
	2	1	0.993	0.011	0.976	0.026	0.984	0.018	0.997	0.003	0.001	0.001	736058
		2	0.997	0.006	0.987	0.017	0.992	0.010	0.999	0.002	0.000	0.001	370871
		3	1.000	0.000	0.992	0.012	0.996	0.006	0.999	0.001	0.000	0.000	281947
	3	1	0.973	0.028	0.957	0.031	0.963	0.025	0.993	0.004	0.003	0.003	985000
		2	0.984	0.019	0.995	0.009	0.989	0.010	0.998	0.002	0.001	0.002	297363
		3	0.997	0.005	0.992	0.012	0.994	0.007	0.999	0.001	0.000	0.000	237816
50	1	1	0.817	0.107	0.695	0.129	0.741	0.115	0.982	0.007	0.007	0.004	3614234
		2	0.916	0.061	0.872	0.052	0.891	0.043	0.992	0.004	0.004	0.003	1704158
		3	0.973	0.031	0.917	0.043	0.943	0.029	0.996	0.002	0.001	0.001	1283640
	2	1	0.941	0.040	0.894	0.048	0.914	0.038	0.993	0.003	0.003	0.002	1881447
		2	0.972	0.028	0.971	0.022	0.971	0.020	0.998	0.002	0.001	0.001	618608
		3	0.987	0.017	0.986	0.016	0.986	0.011	0.999	0.001	0.001	0.001	386190
	3	1	0.977	0.022	0.946	0.039	0.960	0.025	0.997	0.002	0.001	0.001	1386956
		2	0.985	0.018	0.983	0.017	0.984	0.013	0.999	0.001	0.001	0.001	391035
		3	0.994	0.010	0.997	0.005	0.995	0.006	1.000	0.001	0.000	0.000	253289
75	1	1	0.799	0.116	0.661	0.115	0.717	0.110	0.987	0.005	0.005	0.003	5317988
		2	0.930	0.055	0.837	0.084	0.877	0.063	0.994	0.003	0.002	0.001	2330854
		3	0.980	0.026	0.898	0.052	0.935	0.033	0.997	0.001	0.001	0.001	1712962
	2	1	0.940	0.036	0.898	0.051	0.915	0.040	0.995	0.002	0.002	0.001	2404233
		2	0.982	0.021	0.942	0.028	0.961	0.020	0.998	0.001	0.001	0.001	883123
		3	0.973	0.022	0.979	0.019	0.976	0.017	0.999	0.001	0.001	0.001	491412
	3	1	0.978	0.023	0.923	0.027	0.950	0.021	0.997	0.001	0.001	0.001	2415842
		2	0.982	0.018	0.978	0.016	0.980	0.012	0.999	0.001	0.000	0.001	496763
		3	0.993	0.011	0.992	0.011	0.993	0.008	1.000	0.000	0.000	0.000	305689
100	1	1	0.795	0.133	0.618	0.136	0.689	0.134	0.990	0.004	0.003	0.002	6799351
		2	0.938	0.045	0.818	0.078	0.869	0.054	0.995	0.002	0.001	0.001	3031313
		3	0.934	0.048	0.865	0.064	0.894	0.048	0.996	0.002	0.001	0.001	2135500
	2	1	0.945	0.037	0.881	0.053	0.910	0.038	0.996	0.001	0.001	0.001	3386237
		2	0.961	0.032	0.945	0.027	0.953	0.026	0.998	0.001	0.001	0.001	1085567
		3	0.974	0.022	0.959	0.029	0.966	0.018	0.999	0.001	0.001	0.000	696009
	3	1	0.952	0.033	0.910	0.035	0.930	0.028	0.997	0.001	0.001	0.001	3044017
		2	0.972	0.025	0.977	0.014	0.974	0.014	0.999	0.001	0.001	0.001	585360
		3	0.991	0.013	0.989	0.014	0.990	0.010	1.000	0.000	0.000	0.000	374304

Table C.9: PEC results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$f$	$\$c$	PPV		TPR		F_1 score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	1	1	0.958	0.054	0.869	0.101	0.908	0.079	0.986	0.013	0.003	0.004	1810098
		2	0.996	0.008	0.930	0.057	0.960	0.033	0.994	0.005	0.000	0.001	840305
		3	0.934	0.059	0.903	0.054	0.916	0.045	0.984	0.009	0.008	0.007	931409
	2	1	0.988	0.018	0.960	0.038	0.974	0.025	0.995	0.005	0.002	0.002	1026496
		2	0.999	0.001	0.970	0.031	0.984	0.016	0.997	0.003	0.000	0.000	415745
		3	1.000	0.000	0.986	0.017	0.993	0.009	0.999	0.001	0.000	0.000	319222
	3	1	0.943	0.048	0.906	0.067	0.922	0.054	0.982	0.013	0.009	0.008	1722839
		2	0.979	0.022	0.980	0.018	0.979	0.016	0.997	0.003	0.002	0.002	367649
		3	0.994	0.010	0.981	0.016	0.988	0.010	0.998	0.002	0.000	0.001	278117
50	1	1	0.688	0.136	0.493	0.130	0.566	0.127	0.960	0.010	0.013	0.006	5317516
		2	0.871	0.070	0.804	0.067	0.832	0.059	0.984	0.005	0.007	0.004	2246629
		3	0.957	0.039	0.854	0.052	0.901	0.036	0.991	0.003	0.002	0.002	1591080
	2	1	0.890	0.065	0.795	0.107	0.835	0.080	0.982	0.008	0.007	0.004	3496524
		2	0.966	0.030	0.947	0.036	0.956	0.029	0.996	0.003	0.002	0.001	776142
		3	0.991	0.014	0.974	0.023	0.982	0.015	0.998	0.001	0.000	0.001	484009
	3	1	0.933	0.057	0.879	0.103	0.900	0.081	0.989	0.009	0.004	0.004	2753750
		2	0.972	0.024	0.950	0.026	0.960	0.021	0.997	0.002	0.001	0.001	599954
		3	0.986	0.016	0.987	0.015	0.986	0.013	0.999	0.001	0.001	0.001	302731
75	1	1	0.734	0.125	0.500	0.119	0.585	0.115	0.975	0.006	0.007	0.004	7957823
		2	0.874	0.086	0.754	0.088	0.807	0.080	0.989	0.004	0.004	0.002	3168019
		3	0.967	0.037	0.836	0.071	0.894	0.050	0.994	0.003	0.001	0.001	2139999
	2	1	0.863	0.064	0.797	0.092	0.825	0.072	0.987	0.005	0.006	0.003	5134709
		2	0.962	0.036	0.879	0.046	0.916	0.030	0.995	0.002	0.001	0.001	1348029
		3	0.970	0.026	0.943	0.028	0.956	0.022	0.998	0.001	0.001	0.001	687975
	3	1	0.905	0.053	0.788	0.088	0.839	0.067	0.988	0.004	0.004	0.002	6495734
		2	0.967	0.027	0.946	0.031	0.955	0.021	0.998	0.001	0.001	0.001	794895
		3	0.986	0.017	0.982	0.016	0.984	0.013	0.999	0.001	0.000	0.000	366663
100	1	1	0.732	0.133	0.436	0.116	0.539	0.121	0.980	0.004	0.004	0.002	10363800
		2	0.897	0.062	0.754	0.081	0.815	0.068	0.992	0.003	0.002	0.001	4115618
		3	0.909	0.055	0.817	0.053	0.858	0.044	0.994	0.002	0.002	0.001	2876787
	2	1	0.832	0.072	0.729	0.094	0.772	0.073	0.988	0.004	0.005	0.002	7439297
		2	0.938	0.040	0.892	0.039	0.913	0.030	0.996	0.001	0.001	0.001	1652656
		3	0.959	0.023	0.929	0.032	0.943	0.019	0.998	0.001	0.001	0.001	915113
	3	1	0.833	0.068	0.749	0.097	0.784	0.078	0.987	0.004	0.006	0.003	9143203
		2	0.947	0.034	0.935	0.026	0.940	0.022	0.998	0.001	0.001	0.001	1000641
		3	0.983	0.015	0.981	0.016	0.982	0.011	0.999	0.000	0.000	0.000	477340

Table C.10: SBR results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$f$	$\$c$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	1	1	0.196	0.017	1.000	0.000	0.328	0.024	0.819	0.018	0.189	0.018	152000
		2	0.497	0.078	1.000	0.000	0.658	0.070	0.935	0.013	0.068	0.014	77000
		3	0.575	0.069	1.000	0.000	0.726	0.056	0.950	0.012	0.052	0.012	66000
	2	1	0.673	0.066	1.000	0.000	0.802	0.047	0.968	0.007	0.034	0.007	42000
		2	0.907	0.060	1.000	0.000	0.950	0.033	0.992	0.005	0.009	0.006	21000
		3	0.974	0.031	1.000	0.000	0.986	0.016	0.998	0.002	0.002	0.003	26000
	3	1	0.869	0.059	1.000	0.000	0.928	0.034	0.989	0.005	0.011	0.005	32000
		2	0.981	0.025	1.000	0.000	0.990	0.013	0.998	0.002	0.002	0.002	19000
		3	0.993	0.013	1.000	0.000	0.996	0.007	0.999	0.001	0.001	0.001	23000
50	1	1	0.107	0.006	1.000	0.000	0.193	0.010	0.826	0.011	0.178	0.011	346000
		2	0.247	0.032	1.000	0.000	0.395	0.040	0.925	0.009	0.077	0.009	154000
		3	0.329	0.044	1.000	0.000	0.492	0.049	0.949	0.007	0.052	0.007	134000
	2	1	0.466	0.064	1.000	0.000	0.632	0.060	0.966	0.006	0.035	0.006	102000
		2	0.810	0.057	1.000	0.000	0.894	0.035	0.992	0.003	0.008	0.003	37000
		3	0.923	0.055	1.000	0.000	0.958	0.030	0.997	0.002	0.003	0.002	35000
	3	1	0.758	0.066	1.000	0.000	0.860	0.043	0.988	0.003	0.012	0.004	39000
		2	0.962	0.033	1.000	0.000	0.980	0.017	0.998	0.001	0.002	0.001	33000
		3	0.991	0.016	1.000	0.000	0.995	0.008	1.000	0.001	0.000	0.001	30000
75	1	1	0.076	0.006	1.000	0.000	0.141	0.010	0.826	0.010	0.176	0.010	592000
		2	0.164	0.016	1.000	0.000	0.281	0.023	0.925	0.007	0.076	0.007	221000
		3	0.267	0.044	1.000	0.000	0.419	0.055	0.947	0.007	0.053	0.007	165000
	2	1	0.382	0.061	1.000	0.000	0.549	0.063	0.969	0.005	0.032	0.005	113000
		2	0.760	0.078	1.000	0.000	0.860	0.051	0.992	0.003	0.008	0.003	44000
		3	0.877	0.058	1.000	0.000	0.933	0.034	0.996	0.002	0.004	0.002	40000
	3	1	0.667	0.074	1.000	0.000	0.797	0.054	0.988	0.003	0.012	0.003	71000
		2	0.943	0.039	1.000	0.000	0.970	0.021	0.998	0.001	0.002	0.001	43000
		3	0.986	0.020	1.000	0.000	0.993	0.010	1.000	0.001	0.000	0.001	35000
100	1	1	0.059	0.003	1.000	0.000	0.112	0.005	0.835	0.009	0.167	0.009	632000
		2	0.127	0.013	1.000	0.000	0.225	0.020	0.921	0.007	0.079	0.007	329000
		3	0.185	0.020	1.000	0.000	0.311	0.028	0.948	0.005	0.052	0.005	228000
	2	1	0.279	0.038	1.000	0.000	0.434	0.046	0.966	0.005	0.034	0.005	188000
		2	0.684	0.078	1.000	0.000	0.808	0.056	0.992	0.002	0.008	0.002	73000
		3	0.840	0.068	1.000	0.000	0.911	0.041	0.996	0.002	0.004	0.002	65000
	3	1	0.563	0.077	1.000	0.000	0.716	0.063	0.986	0.003	0.014	0.003	101000
		2	0.935	0.046	1.000	0.000	0.965	0.025	0.999	0.001	0.001	0.001	73000
		3	0.975	0.030	1.000	0.000	0.987	0.015	0.999	0.001	0.001	0.001	37000

Table C.11: SBR results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$f$	$\$c$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	1	1	0.200	0.015	1.000	0.000	0.332	0.021	0.738	0.071	0.281	0.080	184243
		2	0.455	0.076	1.000	0.000	0.620	0.071	0.903	0.031	0.103	0.034	92044
		3	0.507	0.059	1.000	0.000	0.669	0.052	0.916	0.016	0.090	0.017	109096
	2	1	0.558	0.132	1.000	0.000	0.705	0.109	0.912	0.058	0.095	0.064	97512
		2	0.814	0.085	1.000	0.000	0.894	0.052	0.979	0.011	0.022	0.011	40690
		3	0.884	0.070	1.000	0.000	0.936	0.040	0.989	0.007	0.012	0.007	34822
	3	1	0.501	0.043	1.000	0.000	0.666	0.038	0.880	0.017	0.130	0.018	146921
		2	0.864	0.044	1.000	0.000	0.926	0.025	0.987	0.005	0.013	0.005	49789
		3	0.917	0.046	1.000	0.000	0.956	0.025	0.993	0.004	0.008	0.004	36529
50	1	1	0.111	0.008	1.000	0.000	0.200	0.012	0.676	0.021	0.337	0.022	489047
		2	0.212	0.026	1.000	0.000	0.349	0.035	0.864	0.013	0.140	0.014	239079
		3	0.282	0.034	1.000	0.000	0.438	0.042	0.908	0.012	0.095	0.013	169249
	2	1	0.245	0.021	1.000	0.000	0.393	0.027	0.843	0.014	0.164	0.014	289000
		2	0.544	0.078	1.000	0.000	0.698	0.063	0.967	0.007	0.034	0.008	91924
		3	0.750	0.077	1.000	0.000	0.854	0.049	0.987	0.005	0.014	0.005	46617
	3	1	0.414	0.145	1.000	0.000	0.564	0.124	0.895	0.037	0.109	0.039	220751
		2	0.741	0.054	1.000	0.000	0.849	0.036	0.985	0.004	0.015	0.004	69614
		3	0.886	0.043	1.000	0.000	0.939	0.024	0.995	0.002	0.005	0.002	62433
75	1	1	0.076	0.005	1.000	0.000	0.141	0.009	0.670	0.018	0.339	0.018	804640
		2	0.149	0.019	1.000	0.000	0.258	0.029	0.862	0.010	0.141	0.010	374813
		3	0.207	0.033	1.000	0.000	0.342	0.045	0.905	0.013	0.097	0.013	239480
	2	1	0.196	0.020	1.000	0.000	0.328	0.028	0.857	0.010	0.147	0.011	389833
		2	0.401	0.045	1.000	0.000	0.570	0.046	0.962	0.005	0.039	0.005	148415
		3	0.625	0.057	1.000	0.000	0.767	0.044	0.984	0.003	0.016	0.003	91018
	3	1	0.238	0.029	1.000	0.000	0.383	0.038	0.859	0.013	0.145	0.014	430211
		2	0.629	0.063	1.000	0.000	0.769	0.048	0.984	0.003	0.016	0.003	90904
		3	0.858	0.043	1.000	0.000	0.923	0.025	0.996	0.001	0.005	0.002	70158
100	1	1	0.060	0.004	1.000	0.000	0.113	0.007	0.682	0.017	0.325	0.017	1030129
		2	0.111	0.015	1.000	0.000	0.200	0.024	0.857	0.011	0.145	0.011	465595
		3	0.156	0.020	1.000	0.000	0.269	0.030	0.906	0.009	0.095	0.009	334137
	2	1	0.143	0.017	1.000	0.000	0.249	0.026	0.846	0.013	0.157	0.013	550904
		2	0.337	0.048	1.000	0.000	0.501	0.053	0.963	0.005	0.038	0.005	192749
		3	0.521	0.056	1.000	0.000	0.683	0.048	0.983	0.003	0.018	0.003	108482
	3	1	0.192	0.030	1.000	0.000	0.321	0.042	0.860	0.014	0.143	0.014	541813
		2	0.575	0.052	1.000	0.000	0.728	0.042	0.986	0.002	0.015	0.002	121246
		3	0.801	0.051	1.000	0.000	0.888	0.032	0.995	0.001	0.005	0.001	73137

Table C.12: SBR results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$f$	$\$c$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
25	1	1	0.203	0.016	1.000	0.000	0.338	0.022	0.700	0.096	0.328	0.112	226287
		2	0.455	0.080	1.000	0.000	0.619	0.074	0.897	0.038	0.111	0.042	106635
		3	0.484	0.051	1.000	0.000	0.650	0.046	0.891	0.018	0.119	0.020	114783
	2	1	0.509	0.130	1.000	0.000	0.662	0.112	0.878	0.077	0.134	0.086	128724
		2	0.738	0.150	1.000	0.000	0.839	0.098	0.963	0.025	0.039	0.027	68210
		3	0.852	0.090	1.000	0.000	0.916	0.053	0.985	0.010	0.016	0.011	40736
	3	1	0.390	0.084	1.000	0.000	0.555	0.085	0.736	0.109	0.298	0.126	272426
		2	0.758	0.073	1.000	0.000	0.860	0.047	0.974	0.010	0.028	0.010	80674
		3	0.829	0.047	1.000	0.000	0.905	0.028	0.984	0.005	0.017	0.005	64638
50	1	1	0.115	0.007	1.000	0.000	0.207	0.011	0.546	0.039	0.481	0.042	891764
		2	0.204	0.023	1.000	0.000	0.338	0.032	0.810	0.016	0.199	0.017	310522
		3	0.258	0.033	1.000	0.000	0.408	0.041	0.870	0.014	0.136	0.015	194558
	2	1	0.188	0.025	1.000	0.000	0.315	0.035	0.685	0.051	0.335	0.054	500171
		2	0.473	0.109	1.000	0.000	0.631	0.098	0.949	0.021	0.053	0.021	117804
		3	0.642	0.094	1.000	0.000	0.775	0.067	0.976	0.008	0.024	0.008	87783
	3	1	0.318	0.133	1.000	0.000	0.454	0.131	0.761	0.124	0.253	0.133	453899
		2	0.513	0.053	1.000	0.000	0.676	0.046	0.961	0.007	0.040	0.007	135702
		3	0.788	0.061	1.000	0.000	0.879	0.038	0.989	0.004	0.011	0.004	90455
75	1	1	0.079	0.005	1.000	0.000	0.146	0.009	0.530	0.030	0.489	0.031	1610252
		2	0.138	0.016	1.000	0.000	0.243	0.025	0.804	0.015	0.202	0.016	475909
		3	0.184	0.030	1.000	0.000	0.310	0.043	0.872	0.018	0.131	0.019	297850
	2	1	0.142	0.015	1.000	0.000	0.248	0.022	0.688	0.030	0.324	0.031	779495
		2	0.272	0.038	1.000	0.000	0.425	0.046	0.925	0.009	0.077	0.009	231772
		3	0.446	0.057	1.000	0.000	0.614	0.055	0.968	0.005	0.032	0.005	135254
	3	1	0.142	0.022	1.000	0.000	0.247	0.034	0.604	0.045	0.413	0.046	1364899
		2	0.419	0.046	1.000	0.000	0.589	0.045	0.961	0.006	0.040	0.006	166658
		3	0.715	0.047	1.000	0.000	0.832	0.032	0.989	0.002	0.011	0.002	95038
100	1	1	0.061	0.004	1.000	0.000	0.116	0.007	0.540	0.027	0.474	0.028	2500439
		2	0.106	0.013	1.000	0.000	0.191	0.021	0.798	0.013	0.206	0.013	612724
		3	0.137	0.016	1.000	0.000	0.240	0.025	0.866	0.011	0.137	0.012	453216
	2	1	0.099	0.013	1.000	0.000	0.179	0.021	0.657	0.032	0.354	0.033	1400142
		2	0.226	0.037	1.000	0.000	0.367	0.049	0.926	0.009	0.075	0.009	279192
		3	0.372	0.046	1.000	0.000	0.540	0.048	0.967	0.004	0.034	0.004	140806
	3	1	0.104	0.016	1.000	0.000	0.188	0.026	0.594	0.046	0.418	0.047	1965743
		2	0.360	0.038	1.000	0.000	0.527	0.042	0.962	0.004	0.039	0.005	192750
		3	0.613	0.055	1.000	0.000	0.758	0.043	0.987	0.003	0.013	0.003	129861

Table C.13: PEC results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$F$	$\$f$	$\$c$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
10	1	1	0.976	0.036	0.847	0.081	0.904	0.052	0.998	0.001	0.000	0.001	3591000
		2	0.982	0.027	0.930	0.056	0.954	0.030	0.999	0.001	0.000	0.000	1834000
		3	0.986	0.022	0.940	0.056	0.961	0.032	0.999	0.001	0.000	0.000	1292000
	2	1	0.979	0.029	0.966	0.050	0.971	0.031	0.999	0.001	0.000	0.001	915000
		2	0.969	0.033	0.978	0.035	0.973	0.025	0.999	0.001	0.001	0.001	361000
		3	0.983	0.023	0.998	0.004	0.990	0.013	1.000	0.000	0.000	0.000	232000
	3	1	0.979	0.031	0.972	0.040	0.975	0.028	0.999	0.001	0.000	0.001	464000
		2	0.993	0.013	0.999	0.002	0.996	0.007	1.000	0.000	0.000	0.000	180000
		3	0.999	0.003	0.999	0.002	0.999	0.002	1.000	0.000	0.000	0.000	178000
20	1	1	0.981	0.028	0.971	0.042	0.975	0.027	0.999	0.001	0.000	0.001	1693000
		2	0.980	0.028	0.932	0.057	0.954	0.031	0.999	0.001	0.000	0.001	1000000
		3	0.996	0.007	0.968	0.046	0.981	0.025	1.000	0.001	0.000	0.000	742000
	2	1	0.986	0.021	0.989	0.020	0.987	0.018	1.000	0.001	0.000	0.000	361000
		2	0.987	0.020	1.000	0.000	0.993	0.010	1.000	0.000	0.000	0.000	207000
		3	0.995	0.010	0.995	0.010	0.995	0.009	1.000	0.000	0.000	0.000	209000
	3	1	0.993	0.012	1.000	0.000	0.996	0.006	1.000	0.000	0.000	0.000	190000
		2	0.999	0.002	1.000	0.000	0.999	0.001	1.000	0.000	0.000	0.000	156000
		3	0.998	0.004	1.000	0.000	0.999	0.002	1.000	0.000	0.000	0.000	160000
30	1	1	0.987	0.021	0.951	0.059	0.967	0.034	0.999	0.001	0.000	0.000	1242000
		2	0.996	0.007	0.975	0.038	0.985	0.022	1.000	0.000	0.000	0.000	733000
		3	0.988	0.019	0.988	0.021	0.988	0.017	1.000	0.000	0.000	0.000	488000
	2	1	0.989	0.017	0.995	0.009	0.992	0.012	1.000	0.000	0.000	0.000	242000
		2	0.998	0.004	1.000	0.000	0.999	0.002	1.000	0.000	0.000	0.000	183000
		3	0.998	0.004	1.000	0.000	0.999	0.002	1.000	0.000	0.000	0.000	189000
	3	1	0.996	0.008	1.000	0.000	0.998	0.004	1.000	0.000	0.000	0.000	163000
		2	0.998	0.005	0.999	0.002	0.998	0.003	1.000	0.000	0.000	0.000	157000
		3	0.997	0.006	1.000	0.000	0.998	0.003	1.000	0.000	0.000	0.000	163000
40	1	1	0.988	0.020	0.962	0.047	0.974	0.025	0.999	0.001	0.000	0.000	1076000
		2	0.995	0.009	0.989	0.020	0.992	0.013	1.000	0.000	0.000	0.000	585000
		3	0.983	0.024	0.977	0.035	0.979	0.022	0.999	0.001	0.000	0.000	466000
	2	1	0.987	0.021	0.999	0.002	0.993	0.011	1.000	0.000	0.000	0.000	189000
		2	0.999	0.003	1.000	0.000	0.999	0.001	1.000	0.000	0.000	0.000	175000
		3	0.999	0.002	1.000	0.000	0.999	0.001	1.000	0.000	0.000	0.000	173000
	3	1	0.993	0.012	1.000	0.000	0.996	0.006	1.000	0.000	0.000	0.000	185000
		2	0.999	0.002	0.999	0.002	0.999	0.002	1.000	0.000	0.000	0.000	168000
		3	0.998	0.004	0.999	0.002	0.998	0.003	1.000	0.000	0.000	0.000	164000
50	1	1	0.969	0.035	0.959	0.052	0.962	0.029	0.999	0.001	0.001	0.001	892000
		2	0.992	0.014	0.983	0.028	0.987	0.019	1.000	0.000	0.000	0.000	471000
		3	0.961	0.030	0.990	0.018	0.975	0.018	0.999	0.001	0.001	0.001	400000
	2	1	0.993	0.012	0.999	0.002	0.996	0.007	1.000	0.000	0.000	0.000	178000
		2	0.999	0.002	1.000	0.000	0.999	0.001	1.000	0.000	0.000	0.000	185000
		3	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	0.000	0.000	170000
	3	1	1.000	0.001	1.000	0.000	1.000	0.001	1.000	0.000	0.000	0.000	171000
		2	0.999	0.002	1.000	0.000	0.999	0.001	1.000	0.000	0.000	0.000	184000
		3	0.999	0.002	1.000	0.000	0.999	0.001	1.000	0.000	0.000	0.000	151000

Table C.14: PEC results for generated simulations based on following parameters: $P=2$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$F$	$\$f$	$\$c$	PPV		TPR		F_1 score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
10	1	1	0.795	0.133	0.618	0.136	0.689	0.134	0.990	0.004	0.003	0.002	6799351
		2	0.938	0.045	0.818	0.078	0.869	0.054	0.995	0.002	0.001	0.001	3031313
		3	0.934	0.048	0.865	0.064	0.894	0.048	0.996	0.002	0.001	0.001	2135500
	2	1	0.945	0.037	0.881	0.053	0.910	0.038	0.996	0.001	0.001	0.001	3386237
		2	0.961	0.032	0.945	0.027	0.953	0.026	0.998	0.001	0.001	0.001	1085567
		3	0.974	0.022	0.959	0.029	0.966	0.018	0.999	0.001	0.001	0.000	696009
	3	1	0.952	0.033	0.910	0.035	0.930	0.028	0.997	0.001	0.001	0.001	3044017
		2	0.972	0.025	0.977	0.014	0.974	0.014	0.999	0.001	0.001	0.001	585360
		3	0.991	0.013	0.989	0.014	0.990	0.010	1.000	0.000	0.000	0.000	374304
20	1	1	0.943	0.051	0.849	0.067	0.889	0.057	0.996	0.002	0.001	0.001	2986211
		2	0.949	0.045	0.888	0.060	0.913	0.047	0.997	0.002	0.001	0.001	1788470
		3	0.986	0.019	0.912	0.047	0.945	0.029	0.998	0.001	0.000	0.000	1352901
	2	1	0.976	0.023	0.926	0.041	0.947	0.030	0.998	0.001	0.001	0.001	1202371
		2	0.977	0.023	0.977	0.015	0.976	0.013	0.999	0.001	0.001	0.001	488096
		3	0.993	0.011	0.985	0.018	0.989	0.010	1.000	0.000	0.000	0.000	384000
	3	1	0.987	0.013	0.977	0.006	0.982	0.008	0.999	0.000	0.000	0.000	784167
		2	0.988	0.015	0.994	0.009	0.991	0.009	1.000	0.000	0.000	0.000	334512
		3	0.992	0.012	0.998	0.003	0.995	0.007	1.000	0.000	0.000	0.000	277251
30	1	1	0.957	0.033	0.856	0.069	0.899	0.048	0.996	0.001	0.001	0.001	2312464
		2	0.981	0.025	0.926	0.036	0.952	0.025	0.998	0.001	0.000	0.001	1315395
		3	0.983	0.022	0.934	0.043	0.956	0.030	0.998	0.001	0.000	0.001	997094
	2	1	0.971	0.023	0.947	0.044	0.954	0.034	0.998	0.001	0.001	0.001	808868
		2	0.992	0.012	0.994	0.009	0.993	0.008	1.000	0.000	0.000	0.000	436389
		3	0.992	0.011	0.995	0.007	0.994	0.007	1.000	0.000	0.000	0.000	349117
	3	1	0.988	0.013	0.990	0.014	0.989	0.009	0.999	0.000	0.000	0.000	538772
		2	0.994	0.009	0.998	0.005	0.996	0.006	1.000	0.000	0.000	0.000	321640
		3	0.992	0.013	0.997	0.006	0.994	0.008	1.000	0.000	0.000	0.000	302474
40	1	1	0.957	0.039	0.909	0.044	0.930	0.037	0.997	0.001	0.001	0.001	1780061
		2	0.987	0.019	0.935	0.034	0.958	0.021	0.998	0.001	0.000	0.001	1085526
		3	0.975	0.026	0.951	0.030	0.962	0.021	0.998	0.001	0.001	0.001	865598
	2	1	0.981	0.017	0.967	0.031	0.972	0.022	0.999	0.001	0.001	0.001	671827
		2	0.991	0.012	0.992	0.011	0.992	0.008	1.000	0.000	0.000	0.000	401474
		3	0.994	0.009	0.998	0.003	0.996	0.006	1.000	0.000	0.000	0.000	359801
	3	1	0.984	0.014	0.985	0.021	0.983	0.015	0.999	0.001	0.000	0.000	509895
		2	0.997	0.006	0.989	0.016	0.993	0.010	1.000	0.000	0.000	0.000	336295
		3	0.997	0.006	0.998	0.004	0.997	0.005	1.000	0.000	0.000	0.000	338404
50	1	1	0.948	0.037	0.900	0.043	0.922	0.030	0.997	0.001	0.001	0.001	1539798
		2	0.987	0.019	0.929	0.041	0.954	0.029	0.998	0.001	0.000	0.000	933026
		3	0.956	0.031	0.959	0.036	0.955	0.025	0.998	0.001	0.001	0.001	774617
	2	1	0.988	0.014	0.982	0.022	0.983	0.015	0.999	0.001	0.000	0.000	529617
		2	0.996	0.006	0.995	0.008	0.996	0.006	1.000	0.000	0.000	0.000	377830
		3	0.999	0.002	0.992	0.014	0.995	0.009	1.000	0.000	0.000	0.000	337316
	3	1	0.991	0.012	0.997	0.006	0.994	0.007	1.000	0.000	0.000	0.000	454763
		2	0.996	0.007	0.996	0.008	0.996	0.007	1.000	0.000	0.000	0.000	386000
		3	0.997	0.005	0.996	0.007	0.997	0.006	1.000	0.000	0.000	0.000	333661

Table C.15: PEC results for generated simulations based on following parameters: $P=3$, $g=100$, $d=11$, $b=1$, $B=2$, $\Phi=1$, $\Psi=0$, $t=1$ and $T=100$

$\$F$	$\$f$	$\$c$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
10	1	1	0.732	0.133	0.436	0.116	0.539	0.121	0.980	0.004	0.004	0.002	10363800
		2	0.897	0.062	0.754	0.081	0.815	0.068	0.992	0.003	0.002	0.001	4115618
		3	0.909	0.055	0.817	0.053	0.858	0.044	0.994	0.002	0.002	0.001	2876787
	2	1	0.832	0.072	0.729	0.094	0.772	0.073	0.988	0.004	0.005	0.002	7439297
		2	0.938	0.040	0.892	0.039	0.913	0.030	0.996	0.001	0.001	0.001	1652656
		3	0.959	0.023	0.929	0.032	0.943	0.019	0.998	0.001	0.001	0.001	915113
	3	1	0.833	0.068	0.749	0.097	0.784	0.078	0.987	0.004	0.006	0.003	9143203
		2	0.947	0.034	0.935	0.026	0.940	0.022	0.998	0.001	0.001	0.001	1000641
		3	0.983	0.015	0.981	0.016	0.982	0.011	0.999	0.000	0.000	0.000	477340
20	1	1	0.924	0.053	0.793	0.101	0.848	0.079	0.993	0.004	0.002	0.001	4122751
		2	0.930	0.040	0.846	0.053	0.883	0.038	0.994	0.002	0.002	0.001	2427415
		3	0.975	0.029	0.874	0.043	0.919	0.031	0.996	0.001	0.001	0.001	1690542
	2	1	0.955	0.025	0.902	0.034	0.927	0.023	0.996	0.001	0.002	0.001	2165578
		2	0.963	0.023	0.946	0.029	0.953	0.021	0.998	0.001	0.001	0.001	742276
		3	0.982	0.018	0.974	0.023	0.978	0.015	0.999	0.001	0.000	0.000	519589
	3	1	0.979	0.016	0.953	0.021	0.965	0.016	0.998	0.001	0.001	0.001	1634092
		2	0.976	0.020	0.982	0.014	0.979	0.012	0.999	0.001	0.001	0.000	424651
		3	0.983	0.016	0.992	0.010	0.987	0.010	0.999	0.000	0.000	0.000	316493
30	1	1	0.936	0.040	0.813	0.081	0.865	0.062	0.993	0.003	0.002	0.001	3196914
		2	0.972	0.028	0.883	0.047	0.923	0.034	0.996	0.002	0.001	0.001	1761252
		3	0.981	0.022	0.885	0.066	0.926	0.044	0.997	0.002	0.001	0.001	1324885
	2	1	0.969	0.020	0.938	0.033	0.951	0.023	0.997	0.001	0.001	0.001	1342273
		2	0.988	0.014	0.984	0.013	0.985	0.010	0.999	0.001	0.000	0.000	571033
		3	0.991	0.011	0.992	0.011	0.991	0.008	1.000	0.000	0.000	0.000	483582
	3	1	0.978	0.015	0.976	0.016	0.976	0.013	0.998	0.001	0.001	0.001	1010318
		2	0.990	0.013	0.992	0.010	0.991	0.009	1.000	0.000	0.000	0.000	392462
		3	0.990	0.012	0.992	0.011	0.991	0.008	1.000	0.000	0.000	0.000	366988
40	1	1	0.955	0.037	0.880	0.054	0.913	0.043	0.996	0.002	0.001	0.001	2527288
		2	0.977	0.022	0.920	0.028	0.947	0.018	0.997	0.001	0.001	0.001	1402450
		3	0.978	0.023	0.933	0.033	0.954	0.022	0.998	0.001	0.001	0.001	1135284
	2	1	0.973	0.019	0.962	0.021	0.967	0.014	0.998	0.001	0.001	0.001	997665
		2	0.986	0.016	0.986	0.015	0.986	0.013	0.999	0.001	0.000	0.000	547549
		3	0.990	0.012	0.991	0.014	0.990	0.011	0.999	0.001	0.000	0.000	468643
	3	1	0.971	0.018	0.985	0.011	0.978	0.011	0.998	0.001	0.001	0.001	833069
		2	0.992	0.012	0.991	0.012	0.991	0.010	1.000	0.000	0.000	0.000	399182
		3	0.995	0.009	0.994	0.010	0.994	0.008	1.000	0.000	0.000	0.000	367770
50	1	1	0.938	0.035	0.862	0.060	0.894	0.044	0.994	0.002	0.002	0.001	2077771
		2	0.985	0.019	0.899	0.062	0.935	0.042	0.997	0.002	0.001	0.001	1252194
		3	0.953	0.029	0.951	0.031	0.951	0.025	0.997	0.001	0.002	0.001	1010258
	2	1	0.984	0.013	0.970	0.026	0.975	0.019	0.999	0.001	0.001	0.000	896138
		2	0.991	0.011	0.978	0.029	0.982	0.019	0.999	0.001	0.000	0.000	530598
		3	0.998	0.004	0.997	0.005	0.997	0.004	1.000	0.000	0.000	0.000	444852
	3	1	0.987	0.012	0.977	0.028	0.980	0.019	0.999	0.001	0.001	0.000	730433
		2	0.994	0.009	0.994	0.010	0.994	0.007	1.000	0.000	0.000	0.000	459701
		3	0.997	0.006	0.998	0.004	0.997	0.004	1.000	0.000	0.000	0.000	383581

Table C.16: SBR results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$F$	$\$f$	$\$c$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
10	1	1	0.059	0.003	1.000	0.000	0.112	0.005	0.835	0.009	0.167	0.009	632000
		2	0.127	0.013	1.000	0.000	0.225	0.020	0.921	0.007	0.079	0.007	329000
		3	0.185	0.020	1.000	0.000	0.311	0.028	0.948	0.005	0.052	0.005	228000
	2	1	0.279	0.038	1.000	0.000	0.434	0.046	0.966	0.005	0.034	0.005	188000
		2	0.684	0.078	1.000	0.000	0.808	0.056	0.992	0.002	0.008	0.002	73000
		3	0.840	0.068	1.000	0.000	0.911	0.041	0.996	0.002	0.004	0.002	65000
	3	1	0.563	0.077	1.000	0.000	0.716	0.063	0.986	0.003	0.014	0.003	101000
		2	0.935	0.046	1.000	0.000	0.965	0.025	0.999	0.001	0.001	0.001	73000
		3	0.975	0.030	1.000	0.000	0.987	0.015	0.999	0.001	0.001	0.001	37000
20	1	1	0.135	0.010	1.000	0.000	0.238	0.015	0.929	0.005	0.071	0.005	316000
		2	0.272	0.045	1.000	0.000	0.424	0.055	0.963	0.007	0.037	0.007	186000
		3	0.356	0.046	1.000	0.000	0.522	0.050	0.975	0.003	0.025	0.003	155000
	2	1	0.706	0.075	1.000	0.000	0.824	0.052	0.992	0.002	0.008	0.002	75000
		2	0.907	0.055	1.000	0.000	0.950	0.030	0.998	0.001	0.002	0.001	47000
		3	0.949	0.036	1.000	0.000	0.973	0.019	0.999	0.001	0.001	0.001	47000
	3	1	0.942	0.039	1.000	0.000	0.969	0.021	0.999	0.001	0.001	0.001	77000
		2	0.990	0.016	1.000	0.000	0.995	0.008	1.000	0.000	0.000	0.000	91000
		3	0.988	0.019	1.000	0.000	0.994	0.010	1.000	0.000	0.000	0.000	37000
30	1	1	0.185	0.018	1.000	0.000	0.312	0.026	0.950	0.005	0.051	0.005	220000
		2	0.358	0.054	1.000	0.000	0.524	0.059	0.974	0.004	0.026	0.004	156000
		3	0.528	0.077	1.000	0.000	0.686	0.065	0.984	0.004	0.016	0.004	97000
	2	1	0.839	0.061	1.000	0.000	0.911	0.036	0.996	0.001	0.004	0.001	64000
		2	0.969	0.031	1.000	0.000	0.984	0.016	0.999	0.001	0.001	0.001	43000
		3	0.978	0.028	1.000	0.000	0.989	0.015	1.000	0.001	0.000	0.001	46000
	3	1	0.979	0.030	1.000	0.000	0.989	0.016	1.000	0.001	0.000	0.001	50000
		2	0.990	0.016	1.000	0.000	0.995	0.008	1.000	0.000	0.000	0.000	62000
		3	0.975	0.031	1.000	0.000	0.987	0.016	0.999	0.001	0.001	0.001	63000
40	1	1	0.244	0.035	1.000	0.000	0.390	0.045	0.960	0.005	0.041	0.005	175000
		2	0.462	0.061	1.000	0.000	0.628	0.057	0.982	0.003	0.018	0.003	104000
		3	0.580	0.074	1.000	0.000	0.730	0.060	0.988	0.003	0.012	0.003	81000
	2	1	0.899	0.043	1.000	0.000	0.946	0.024	0.998	0.001	0.002	0.001	66000
		2	0.980	0.025	1.000	0.000	0.990	0.013	1.000	0.000	0.000	0.001	56000
		3	0.991	0.015	1.000	0.000	0.995	0.008	1.000	0.000	0.000	0.000	91000
	3	1	0.978	0.027	1.000	0.000	0.989	0.014	1.000	0.001	0.000	0.001	88000
		2	0.976	0.030	1.000	0.000	0.987	0.016	1.000	0.001	0.000	0.001	61000
		3	0.960	0.035	1.000	0.000	0.979	0.018	0.999	0.001	0.001	0.001	83000
50	1	1	0.324	0.051	1.000	0.000	0.486	0.058	0.969	0.004	0.031	0.004	173000
		2	0.565	0.081	1.000	0.000	0.717	0.067	0.986	0.003	0.014	0.003	118000
		3	0.629	0.075	1.000	0.000	0.768	0.056	0.990	0.003	0.010	0.003	90000
	2	1	0.940	0.048	1.000	0.000	0.968	0.026	0.999	0.001	0.001	0.001	70000
		2	0.985	0.021	1.000	0.000	0.992	0.011	1.000	0.000	0.000	0.000	55000
		3	0.991	0.015	1.000	0.000	0.995	0.008	1.000	0.000	0.000	0.000	56000
	3	1	0.991	0.015	1.000	0.000	0.995	0.008	1.000	0.000	0.000	0.000	66000
		2	0.987	0.019	1.000	0.000	0.993	0.010	1.000	0.000	0.000	0.000	68000
		3	0.985	0.022	1.000	0.000	0.992	0.012	1.000	0.000	0.000	0.000	58000

Table C.17: SBR results for generated simulations based on following parameters: $P=2$, $g=100$, $d=11$, $b=1$, $B=2$, $\Phi=1$, $\Psi=0$, $t=1$ and $T=100$

F	f	c	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
10	1	1	0.060	0.004	1.000	0.000	0.113	0.007	0.682	0.017	0.325	0.017	1030129
		2	0.111	0.015	1.000	0.000	0.200	0.024	0.857	0.011	0.145	0.011	465595
		3	0.156	0.020	1.000	0.000	0.269	0.030	0.906	0.009	0.095	0.009	334137
	2	1	0.143	0.017	1.000	0.000	0.249	0.026	0.846	0.013	0.157	0.013	550904
		2	0.337	0.048	1.000	0.000	0.501	0.053	0.963	0.005	0.038	0.005	192749
		3	0.521	0.056	1.000	0.000	0.683	0.048	0.983	0.003	0.018	0.003	108482
	3	1	0.192	0.030	1.000	0.000	0.321	0.042	0.860	0.014	0.143	0.014	541813
		2	0.575	0.052	1.000	0.000	0.728	0.042	0.986	0.002	0.015	0.002	121246
		3	0.801	0.051	1.000	0.000	0.888	0.032	0.995	0.001	0.005	0.001	73137
20	1	1	0.129	0.009	1.000	0.000	0.229	0.014	0.863	0.010	0.139	0.010	487158
		2	0.228	0.029	1.000	0.000	0.370	0.038	0.931	0.008	0.071	0.008	256481
		3	0.294	0.038	1.000	0.000	0.452	0.046	0.953	0.005	0.048	0.006	178863
	2	1	0.405	0.042	1.000	0.000	0.574	0.042	0.961	0.004	0.039	0.005	194137
		2	0.722	0.060	1.000	0.000	0.837	0.040	0.991	0.002	0.009	0.002	90430
		3	0.823	0.054	1.000	0.000	0.901	0.033	0.995	0.002	0.005	0.002	88737
	3	1	0.623	0.042	1.000	0.000	0.766	0.032	0.983	0.003	0.018	0.003	151424
		2	0.901	0.039	1.000	0.000	0.947	0.022	0.998	0.001	0.002	0.001	96304
		3	0.946	0.033	1.000	0.000	0.972	0.018	0.999	0.001	0.001	0.001	73237
30	1	1	0.178	0.013	1.000	0.000	0.302	0.019	0.904	0.007	0.098	0.007	339161
		2	0.317	0.035	1.000	0.000	0.479	0.040	0.952	0.006	0.049	0.006	191395
		3	0.433	0.047	1.000	0.000	0.602	0.046	0.970	0.004	0.030	0.004	150357
	2	1	0.592	0.046	1.000	0.000	0.742	0.036	0.981	0.003	0.019	0.003	133053
		2	0.850	0.039	1.000	0.000	0.918	0.023	0.996	0.001	0.004	0.001	95094
		3	0.919	0.038	1.000	0.000	0.957	0.021	0.998	0.001	0.002	0.001	76161
	3	1	0.833	0.041	1.000	0.000	0.908	0.024	0.994	0.002	0.006	0.002	147904
		2	0.955	0.030	1.000	0.000	0.977	0.016	0.999	0.001	0.001	0.001	96213
		3	0.928	0.039	1.000	0.000	0.962	0.021	0.998	0.001	0.002	0.001	76079
40	1	1	0.233	0.022	1.000	0.000	0.377	0.028	0.929	0.007	0.072	0.007	243649
		2	0.382	0.044	1.000	0.000	0.551	0.046	0.966	0.005	0.035	0.005	155117
		3	0.493	0.048	1.000	0.000	0.658	0.044	0.977	0.004	0.024	0.004	122475
	2	1	0.722	0.046	1.000	0.000	0.837	0.031	0.989	0.002	0.011	0.002	131713
		2	0.912	0.035	1.000	0.000	0.954	0.019	0.998	0.001	0.002	0.001	97711
		3	0.959	0.026	1.000	0.000	0.979	0.014	0.999	0.001	0.001	0.001	91453
	3	1	0.894	0.033	1.000	0.000	0.944	0.019	0.997	0.001	0.003	0.001	159526
		2	0.942	0.035	1.000	0.000	0.969	0.019	0.999	0.001	0.001	0.001	101930
		3	0.916	0.047	1.000	0.000	0.955	0.026	0.998	0.001	0.002	0.001	81737
50	1	1	0.282	0.030	1.000	0.000	0.439	0.037	0.942	0.007	0.059	0.007	227231
		2	0.497	0.051	1.000	0.000	0.661	0.046	0.975	0.004	0.026	0.004	124874
		3	0.574	0.058	1.000	0.000	0.726	0.046	0.982	0.003	0.018	0.003	121114
	2	1	0.820	0.044	1.000	0.000	0.900	0.027	0.994	0.002	0.006	0.002	119281
		2	0.947	0.027	1.000	0.000	0.973	0.014	0.999	0.001	0.001	0.001	95135
		3	0.974	0.020	1.000	0.000	0.987	0.010	0.999	0.001	0.001	0.001	91526
	3	1	0.936	0.030	1.000	0.000	0.966	0.016	0.998	0.001	0.002	0.001	151687
		2	0.957	0.028	1.000	0.000	0.978	0.015	0.999	0.001	0.001	0.001	81579
		3	0.959	0.032	1.000	0.000	0.978	0.017	0.999	0.001	0.001	0.001	73249

Table C.18: SBR results for generated simulations based on following parameters: $P=3$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$F$	$\$f$	$\$c$	PPV		TPR		F ₁ score		ACC		FPR		Time (ns)
			Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	Mean	MAD	
10	1	1	0.061	0.004	1.000	0.000	0.116	0.007	0.540	0.027	0.474	0.028	2500439
		2	0.106	0.013	1.000	0.000	0.191	0.021	0.798	0.013	0.206	0.013	612724
		3	0.137	0.016	1.000	0.000	0.240	0.025	0.866	0.011	0.137	0.012	453216
	2	1	0.099	0.013	1.000	0.000	0.179	0.021	0.657	0.032	0.354	0.033	1400142
		2	0.226	0.037	1.000	0.000	0.367	0.049	0.926	0.009	0.075	0.009	279192
		3	0.372	0.046	1.000	0.000	0.540	0.048	0.967	0.004	0.034	0.004	140806
	3	1	0.104	0.016	1.000	0.000	0.188	0.026	0.594	0.046	0.418	0.047	1965743
		2	0.360	0.038	1.000	0.000	0.527	0.042	0.962	0.004	0.039	0.005	192750
		3	0.613	0.055	1.000	0.000	0.758	0.043	0.987	0.003	0.013	0.003	129861
20	1	1	0.129	0.008	1.000	0.000	0.229	0.012	0.801	0.017	0.205	0.017	633594
		2	0.209	0.027	1.000	0.000	0.345	0.037	0.898	0.010	0.105	0.011	357626
		3	0.274	0.034	1.000	0.000	0.428	0.041	0.932	0.007	0.070	0.007	246164
	2	1	0.294	0.027	1.000	0.000	0.453	0.032	0.913	0.009	0.090	0.009	346963
		2	0.548	0.059	1.000	0.000	0.705	0.050	0.980	0.003	0.020	0.003	134834
		3	0.704	0.057	1.000	0.000	0.825	0.040	0.990	0.002	0.010	0.002	112840
	3	1	0.419	0.033	1.000	0.000	0.590	0.033	0.942	0.007	0.060	0.007	349813
		2	0.747	0.051	1.000	0.000	0.853	0.034	0.993	0.002	0.008	0.002	175968
		3	0.867	0.042	1.000	0.000	0.928	0.024	0.997	0.001	0.003	0.001	130389
30	1	1	0.179	0.013	1.000	0.000	0.304	0.019	0.859	0.010	0.145	0.010	456578
		2	0.293	0.032	1.000	0.000	0.452	0.038	0.931	0.008	0.070	0.008	243112
		3	0.382	0.050	1.000	0.000	0.550	0.052	0.955	0.006	0.046	0.006	182568
	2	1	0.483	0.033	1.000	0.000	0.650	0.030	0.960	0.005	0.041	0.005	215013
		2	0.744	0.046	1.000	0.000	0.852	0.030	0.991	0.002	0.009	0.002	127754
		3	0.852	0.045	1.000	0.000	0.919	0.027	0.996	0.001	0.004	0.001	105377
	3	1	0.653	0.029	1.000	0.000	0.790	0.021	0.980	0.002	0.021	0.002	322521
		2	0.878	0.041	1.000	0.000	0.934	0.024	0.997	0.001	0.003	0.001	143316
		3	0.868	0.044	1.000	0.000	0.929	0.025	0.997	0.001	0.003	0.001	115527
40	1	1	0.225	0.016	1.000	0.000	0.367	0.022	0.894	0.009	0.109	0.009	305684
		2	0.378	0.037	1.000	0.000	0.547	0.039	0.952	0.006	0.049	0.007	178231
		3	0.466	0.047	1.000	0.000	0.634	0.044	0.967	0.004	0.034	0.004	154124
	2	1	0.618	0.041	1.000	0.000	0.763	0.031	0.977	0.003	0.024	0.004	196520
		2	0.844	0.040	1.000	0.000	0.914	0.024	0.995	0.001	0.005	0.001	136851
		3	0.919	0.030	1.000	0.000	0.957	0.016	0.998	0.001	0.002	0.001	123465
	3	1	0.782	0.033	1.000	0.000	0.877	0.021	0.989	0.002	0.011	0.002	307400
		2	0.897	0.044	1.000	0.000	0.945	0.025	0.997	0.001	0.003	0.001	123477
		3	0.858	0.052	1.000	0.000	0.922	0.031	0.996	0.001	0.004	0.001	109212
50	1	1	0.281	0.022	1.000	0.000	0.438	0.027	0.918	0.007	0.084	0.008	274620
		2	0.455	0.042	1.000	0.000	0.624	0.040	0.963	0.005	0.038	0.005	161793
		3	0.531	0.053	1.000	0.000	0.691	0.046	0.974	0.004	0.027	0.004	139791
	2	1	0.724	0.039	1.000	0.000	0.839	0.026	0.985	0.003	0.015	0.003	178608
		2	0.889	0.033	1.000	0.000	0.941	0.019	0.996	0.001	0.004	0.001	138155
		3	0.946	0.029	1.000	0.000	0.972	0.015	0.998	0.001	0.002	0.001	129759
	3	1	0.859	0.032	1.000	0.000	0.923	0.018	0.994	0.002	0.007	0.002	264337
		2	0.897	0.038	1.000	0.000	0.945	0.022	0.997	0.001	0.003	0.001	135416
		3	0.928	0.039	1.000	0.000	0.962	0.021	0.998	0.001	0.002	0.001	104510

Table C.19: PEC all results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$F$	$\$f$	$\$c$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
10	1	1	0.976	0.847	0.904	0.998	0.002	0.836	0.000	1.000	0.153	0.998	0.024	0.002	3591000
		2	0.982	0.930	0.954	0.999	0.001	0.921	0.000	1.000	0.070	0.999	0.018	0.001	1834000
		3	0.986	0.940	0.961	0.999	0.001	0.932	0.000	1.000	0.060	0.999	0.014	0.001	1292000
	2	1	0.979	0.966	0.971	0.999	0.001	0.954	0.000	1.000	0.034	1.000	0.021	0.000	915000
		2	0.969	0.978	0.973	0.999	0.001	0.961	0.001	0.999	0.022	1.000	0.031	0.000	361000
		3	0.983	0.998	0.990	1.000	0.000	0.988	0.000	1.000	0.002	1.000	0.017	0.000	232000
	3	1	0.979	0.972	0.975	0.999	0.001	0.960	0.000	1.000	0.028	1.000	0.021	0.000	464000
		2	0.993	0.999	0.996	1.000	0.000	0.995	0.000	1.000	0.001	1.000	0.008	0.000	180000
		3	0.999	0.999	0.999	1.000	0.000	0.998	0.000	1.000	0.001	1.000	0.002	0.000	178000
20	1	1	0.981	0.971	0.975	0.999	0.001	0.960	0.000	1.000	0.029	1.000	0.019	0.000	1693000
		2	0.980	0.932	0.954	0.999	0.001	0.921	0.000	1.000	0.068	0.999	0.020	0.001	1000000
		3	0.996	0.968	0.981	1.000	0.000	0.966	0.000	1.000	0.032	1.000	0.004	0.000	742000
	2	1	0.986	0.989	0.987	1.000	0.000	0.982	0.000	1.000	0.011	1.000	0.014	0.000	361000
		2	0.987	1.000	0.993	1.000	0.000	0.992	0.000	1.000	0.000	1.000	0.013	0.000	207000
		3	0.995	0.995	0.995	1.000	0.000	0.992	0.000	1.000	0.005	1.000	0.005	0.000	209000
	3	1	0.993	1.000	0.996	1.000	0.000	0.996	0.000	1.000	0.000	1.000	0.007	0.000	190000
		2	0.999	1.000	0.999	1.000	0.000	0.999	0.000	1.000	0.000	1.000	0.001	0.000	156000
		3	0.998	1.000	0.999	1.000	0.000	0.999	0.000	1.000	0.000	1.000	0.002	0.000	160000
30	1	1	0.987	0.951	0.967	0.999	0.001	0.944	0.000	1.000	0.049	1.000	0.013	0.000	1242000
		2	0.996	0.975	0.985	1.000	0.000	0.973	0.000	1.000	0.025	1.000	0.004	0.000	733000
		3	0.988	0.988	0.988	1.000	0.000	0.981	0.000	1.000	0.012	1.000	0.012	0.000	488000
	2	1	0.989	0.995	0.992	1.000	0.000	0.989	0.000	1.000	0.005	1.000	0.011	0.000	242000
		2	0.998	1.000	0.999	1.000	0.000	0.999	0.000	1.000	0.000	1.000	0.002	0.000	183000
		3	0.998	1.000	0.999	1.000	0.000	0.999	0.000	1.000	0.000	1.000	0.002	0.000	189000
	3	1	0.996	1.000	0.998	1.000	0.000	0.997	0.000	1.000	0.000	1.000	0.005	0.000	163000
		2	0.998	0.999	0.998	1.000	0.000	0.998	0.000	1.000	0.001	1.000	0.003	0.000	157000
		3	0.997	1.000	0.998	1.000	0.000	0.998	0.000	1.000	0.000	1.000	0.003	0.000	163000
40	1	1	0.988	0.962	0.974	0.999	0.001	0.955	0.000	1.000	0.038	1.000	0.013	0.000	1076000
		2	0.995	0.989	0.992	1.000	0.000	0.986	0.000	1.000	0.011	1.000	0.005	0.000	585000
		3	0.983	0.977	0.979	0.999	0.001	0.967	0.000	1.000	0.023	1.000	0.017	0.000	466000
	2	1	0.987	0.999	0.993	1.000	0.000	0.991	0.000	1.000	0.001	1.000	0.014	0.000	189000
		2	0.999	1.000	0.999	1.000	0.000	0.999	0.000	1.000	0.000	1.000	0.002	0.000	175000
		3	0.999	1.000	0.999	1.000	0.000	0.999	0.000	1.000	0.000	1.000	0.001	0.000	173000
	3	1	0.993	1.000	0.996	1.000	0.000	0.996	0.000	1.000	0.000	1.000	0.007	0.000	185000
		2	0.999	0.999	0.999	1.000	0.000	0.998	0.000	1.000	0.001	1.000	0.001	0.000	168000
		3	0.998	0.999	0.998	1.000	0.000	0.998	0.000	1.000	0.001	1.000	0.002	0.000	164000
50	1	1	0.969	0.959	0.962	0.999	0.001	0.941	0.001	0.999	0.041	1.000	0.031	0.000	892000
		2	0.992	0.983	0.987	1.000	0.000	0.978	0.000	1.000	0.017	1.000	0.008	0.000	471000
		3	0.961	0.990	0.975	0.999	0.001	0.967	0.001	0.999	0.010	1.000	0.039	0.000	400000
	2	1	0.993	0.999	0.996	1.000	0.000	0.995	0.000	1.000	0.001	1.000	0.007	0.000	178000
		2	0.999	1.000	0.999	1.000	0.000	0.999	0.000	1.000	0.000	1.000	0.001	0.000	185000
		3	1.000	1.000	1.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	0.000	170000
	3	1	1.000	1.000	1.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.001	0.000	171000
		2	0.999	1.000	0.999	1.000	0.000	0.999	0.000	1.000	0.000	1.000	0.001	0.000	184000
		3	0.999	1.000	0.999	1.000	0.000	0.999	0.000	1.000	0.000	1.000	0.001	0.000	151000

Table C.20: PEC all results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$F$	$\$f$	$\$c$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
10	1	1	0.795	0.618	0.689	0.990	0.010	0.639	0.003	0.997	0.382	0.993	0.205	0.007	6799351
		2	0.938	0.818	0.869	0.995	0.005	0.830	0.001	0.999	0.182	0.997	0.062	0.003	3031313
		3	0.934	0.865	0.894	0.996	0.004	0.863	0.001	0.999	0.135	0.998	0.066	0.002	2135500
	2	1	0.945	0.881	0.910	0.996	0.004	0.885	0.001	0.999	0.119	0.998	0.055	0.002	3386237
		2	0.961	0.945	0.953	0.998	0.002	0.938	0.001	0.999	0.055	0.999	0.039	0.001	1085567
		3	0.974	0.959	0.966	0.999	0.001	0.955	0.001	0.999	0.041	0.999	0.026	0.001	696009
	3	1	0.952	0.910	0.930	0.997	0.003	0.907	0.001	0.999	0.090	0.998	0.048	0.002	3044017
		2	0.972	0.977	0.974	0.999	0.001	0.968	0.001	0.999	0.023	1.000	0.028	0.000	585360
		3	0.991	0.989	0.990	1.000	0.000	0.987	0.000	1.000	0.011	1.000	0.009	0.000	374304
20	1	1	0.943	0.849	0.889	0.996	0.004	0.860	0.001	0.999	0.151	0.997	0.057	0.003	2986211
		2	0.949	0.888	0.913	0.997	0.003	0.887	0.001	0.999	0.112	0.998	0.051	0.002	1788470
		3	0.986	0.912	0.945	0.998	0.002	0.925	0.000	1.000	0.088	0.998	0.014	0.002	1352901
	2	1	0.976	0.926	0.947	0.998	0.002	0.932	0.001	0.999	0.074	0.999	0.024	0.001	1202371
		2	0.977	0.977	0.976	0.999	0.001	0.971	0.001	0.999	0.023	1.000	0.023	0.000	488096
		3	0.993	0.985	0.989	1.000	0.000	0.986	0.000	1.000	0.015	1.000	0.007	0.000	384000
	3	1	0.987	0.977	0.982	0.999	0.001	0.979	0.000	1.000	0.023	1.000	0.013	0.000	784167
		2	0.988	0.994	0.991	1.000	0.000	0.989	0.000	1.000	0.006	1.000	0.012	0.000	334512
		3	0.992	0.998	0.995	1.000	0.000	0.994	0.000	1.000	0.002	1.000	0.008	0.000	277251
30	1	1	0.957	0.856	0.899	0.996	0.004	0.865	0.001	0.999	0.144	0.997	0.043	0.003	2312464
		2	0.981	0.926	0.952	0.998	0.002	0.934	0.000	1.000	0.074	0.999	0.019	0.001	1315395
		3	0.983	0.934	0.956	0.998	0.002	0.943	0.000	1.000	0.066	0.999	0.017	0.001	997094
	2	1	0.971	0.947	0.954	0.998	0.002	0.945	0.001	0.999	0.053	0.999	0.029	0.001	808868
		2	0.992	0.994	0.993	1.000	0.000	0.992	0.000	1.000	0.006	1.000	0.008	0.000	436389
		3	0.992	0.995	0.994	1.000	0.000	0.993	0.000	1.000	0.005	1.000	0.008	0.000	349117
	3	1	0.988	0.990	0.989	0.999	0.001	0.987	0.000	1.000	0.010	1.000	0.012	0.000	538772
		2	0.994	0.998	0.996	1.000	0.000	0.995	0.000	1.000	0.002	1.000	0.006	0.000	321640
		3	0.992	0.997	0.994	1.000	0.000	0.993	0.000	1.000	0.003	1.000	0.008	0.000	302474
40	1	1	0.957	0.909	0.930	0.997	0.003	0.909	0.001	0.999	0.091	0.998	0.043	0.002	1780061
		2	0.987	0.935	0.958	0.998	0.002	0.944	0.000	1.000	0.065	0.999	0.013	0.001	1085526
		3	0.975	0.951	0.962	0.998	0.002	0.951	0.001	0.999	0.049	0.999	0.025	0.001	865598
	2	1	0.981	0.967	0.972	0.999	0.001	0.965	0.001	0.999	0.033	0.999	0.019	0.001	671827
		2	0.991	0.992	0.992	1.000	0.000	0.991	0.000	1.000	0.008	1.000	0.009	0.000	401474
		3	0.994	0.998	0.996	1.000	0.000	0.996	0.000	1.000	0.002	1.000	0.006	0.000	359801
	3	1	0.984	0.985	0.983	0.999	0.001	0.981	0.000	1.000	0.015	1.000	0.016	0.000	509895
		2	0.997	0.989	0.993	1.000	0.000	0.991	0.000	1.000	0.011	1.000	0.003	0.000	336295
		3	0.997	0.998	0.997	1.000	0.000	0.997	0.000	1.000	0.002	1.000	0.003	0.000	338404
50	1	1	0.948	0.900	0.922	0.997	0.003	0.894	0.001	0.999	0.100	0.998	0.052	0.002	1539798
		2	0.987	0.929	0.954	0.998	0.002	0.939	0.000	1.000	0.071	0.999	0.013	0.001	933026
		3	0.956	0.959	0.955	0.998	0.002	0.947	0.001	0.999	0.041	0.999	0.044	0.001	774617
	2	1	0.988	0.982	0.983	0.999	0.001	0.981	0.000	1.000	0.018	1.000	0.012	0.000	529617
		2	0.996	0.995	0.996	1.000	0.000	0.995	0.000	1.000	0.005	1.000	0.004	0.000	377830
		3	0.999	0.992	0.995	1.000	0.000	0.994	0.000	1.000	0.008	1.000	0.001	0.000	337316
	3	1	0.991	0.997	0.994	1.000	0.000	0.993	0.000	1.000	0.003	1.000	0.009	0.000	454763
		2	0.996	0.996	0.996	1.000	0.000	0.994	0.000	1.000	0.004	1.000	0.004	0.000	386000
		3	0.997	0.996	0.997	1.000	0.000	0.996	0.000	1.000	0.004	1.000	0.003	0.000	333661

Table C.21: PEC all results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$F$	$\$f$	$\$c$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
10	1	1	0.732	0.436	0.539	0.980	0.020	0.495	0.004	0.996	0.564	0.984	0.268	0.016	10363800
		2	0.897	0.754	0.815	0.992	0.008	0.781	0.002	0.998	0.246	0.994	0.103	0.006	4115618
		3	0.909	0.817	0.858	0.994	0.006	0.826	0.002	0.998	0.183	0.996	0.091	0.004	2876787
	2	1	0.832	0.729	0.772	0.988	0.012	0.739	0.005	0.995	0.271	0.992	0.168	0.008	7439297
		2	0.938	0.892	0.913	0.996	0.004	0.894	0.001	0.999	0.108	0.998	0.062	0.002	1652656
		3	0.959	0.929	0.943	0.998	0.002	0.929	0.001	0.999	0.071	0.999	0.041	0.001	915113
	3	1	0.833	0.749	0.784	0.987	0.013	0.755	0.006	0.994	0.251	0.993	0.167	0.007	9143203
		2	0.947	0.935	0.940	0.998	0.002	0.928	0.001	0.999	0.065	0.999	0.053	0.001	1000641
		3	0.983	0.981	0.982	0.999	0.001	0.978	0.000	1.000	0.019	1.000	0.017	0.000	477340
20	1	1	0.924	0.793	0.848	0.993	0.007	0.823	0.002	0.998	0.207	0.994	0.076	0.006	4122751
		2	0.930	0.846	0.883	0.994	0.006	0.857	0.002	0.998	0.154	0.996	0.070	0.004	2427415
		3	0.975	0.874	0.919	0.996	0.004	0.896	0.001	0.999	0.126	0.997	0.025	0.003	1690542
	2	1	0.955	0.902	0.927	0.996	0.004	0.910	0.002	0.998	0.098	0.997	0.045	0.003	2165578
		2	0.963	0.946	0.953	0.998	0.002	0.944	0.001	0.999	0.054	0.999	0.037	0.001	742276
		3	0.982	0.974	0.978	0.999	0.001	0.973	0.000	1.000	0.026	0.999	0.018	0.001	519589
	3	1	0.979	0.953	0.965	0.998	0.002	0.961	0.001	0.999	0.047	0.999	0.021	0.001	1634092
		2	0.976	0.982	0.979	0.999	0.001	0.975	0.001	0.999	0.018	1.000	0.024	0.000	424651
		3	0.983	0.992	0.987	0.999	0.001	0.985	0.000	1.000	0.008	1.000	0.017	0.000	316493
30	1	1	0.936	0.813	0.865	0.993	0.007	0.838	0.002	0.998	0.187	0.995	0.064	0.005	3196914
		2	0.972	0.883	0.923	0.996	0.004	0.905	0.001	0.999	0.117	0.997	0.028	0.003	1761252
		3	0.981	0.885	0.926	0.997	0.003	0.911	0.001	0.999	0.115	0.997	0.019	0.003	1324885
	2	1	0.969	0.938	0.951	0.997	0.003	0.942	0.001	0.999	0.062	0.998	0.031	0.002	1342273
		2	0.988	0.984	0.985	0.999	0.001	0.983	0.000	1.000	0.016	1.000	0.012	0.000	571033
		3	0.991	0.992	0.991	1.000	0.000	0.990	0.000	1.000	0.008	1.000	0.009	0.000	483582
	3	1	0.978	0.976	0.976	0.998	0.002	0.974	0.001	0.999	0.024	0.999	0.022	0.001	1010318
		2	0.990	0.992	0.991	1.000	0.000	0.989	0.000	1.000	0.008	1.000	0.010	0.000	392462
		3	0.990	0.992	0.991	1.000	0.000	0.989	0.000	1.000	0.008	1.000	0.010	0.000	366988
40	1	1	0.955	0.880	0.913	0.996	0.004	0.894	0.001	0.999	0.120	0.997	0.045	0.003	2527288
		2	0.977	0.920	0.947	0.997	0.003	0.933	0.001	0.999	0.080	0.998	0.023	0.002	1402450
		3	0.978	0.933	0.954	0.998	0.002	0.942	0.001	0.999	0.067	0.998	0.022	0.002	1135284
	2	1	0.973	0.962	0.967	0.998	0.002	0.961	0.001	0.999	0.038	0.999	0.027	0.001	997665
		2	0.986	0.986	0.986	0.999	0.001	0.984	0.000	1.000	0.014	1.000	0.014	0.000	547549
		3	0.990	0.991	0.990	0.999	0.001	0.988	0.000	1.000	0.009	1.000	0.010	0.000	468643
	3	1	0.971	0.985	0.978	0.998	0.002	0.975	0.001	0.999	0.015	1.000	0.029	0.000	833069
		2	0.992	0.991	0.991	1.000	0.000	0.990	0.000	1.000	0.009	1.000	0.008	0.000	399182
		3	0.995	0.994	0.994	1.000	0.000	0.993	0.000	1.000	0.006	1.000	0.005	0.000	367770
50	1	1	0.938	0.862	0.894	0.994	0.006	0.871	0.002	0.998	0.138	0.996	0.062	0.004	2077771
		2	0.985	0.899	0.935	0.997	0.003	0.922	0.001	0.999	0.101	0.997	0.015	0.003	1252194
		3	0.953	0.951	0.951	0.997	0.003	0.942	0.002	0.998	0.049	0.999	0.047	0.001	1010258
	2	1	0.984	0.970	0.975	0.999	0.001	0.972	0.001	0.999	0.030	0.999	0.016	0.001	896138
		2	0.991	0.978	0.982	0.999	0.001	0.980	0.000	1.000	0.022	0.999	0.009	0.001	530598
		3	0.998	0.997	0.997	1.000	0.000	0.997	0.000	1.000	0.003	1.000	0.002	0.000	444852
	3	1	0.987	0.977	0.980	0.999	0.001	0.979	0.001	0.999	0.023	0.999	0.013	0.001	730433
		2	0.994	0.994	0.994	1.000	0.000	0.993	0.000	1.000	0.006	1.000	0.006	0.000	459701
		3	0.997	0.998	0.997	1.000	0.000	0.997	0.000	1.000	0.002	1.000	0.003	0.000	383581

Table C.22: SBR all results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$F$	$\$f$	$\$c$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
10	1	1	0.059	1.000	0.112	0.835	0.165	0.221	0.167	0.833	0.000	1.000	0.941	0.000	632000
		2	0.127	1.000	0.225	0.921	0.079	0.337	0.079	0.921	0.000	1.000	0.873	0.000	329000
		3	0.185	1.000	0.311	0.948	0.052	0.412	0.052	0.948	0.000	1.000	0.815	0.000	228000
	2	1	0.279	1.000	0.434	0.966	0.034	0.506	0.034	0.966	0.000	1.000	0.721	0.000	188000
		2	0.684	1.000	0.808	0.992	0.008	0.804	0.008	0.992	0.000	1.000	0.316	0.000	73000
		3	0.840	1.000	0.911	0.996	0.004	0.904	0.004	0.996	0.000	1.000	0.160	0.000	65000
	3	1	0.563	1.000	0.716	0.986	0.014	0.720	0.014	0.986	0.000	1.000	0.437	0.000	101000
		2	0.935	1.000	0.965	0.999	0.001	0.961	0.001	0.999	0.000	1.000	0.065	0.000	73000
		3	0.975	1.000	0.987	0.999	0.001	0.985	0.001	0.999	0.000	1.000	0.026	0.000	37000
20	1	1	0.135	1.000	0.238	0.929	0.071	0.351	0.071	0.929	0.000	1.000	0.865	0.000	316000
		2	0.272	1.000	0.424	0.963	0.037	0.496	0.037	0.963	0.000	1.000	0.728	0.000	186000
		3	0.356	1.000	0.522	0.975	0.025	0.572	0.025	0.975	0.000	1.000	0.644	0.000	155000
	2	1	0.706	1.000	0.824	0.992	0.008	0.817	0.008	0.992	0.000	1.000	0.294	0.000	75000
		2	0.907	1.000	0.950	0.998	0.002	0.944	0.002	0.998	0.000	1.000	0.093	0.000	47000
		3	0.949	1.000	0.973	0.999	0.001	0.969	0.001	0.999	0.000	1.000	0.051	0.000	47000
	3	1	0.942	1.000	0.969	0.999	0.001	0.965	0.001	0.999	0.000	1.000	0.058	0.000	77000
		2	0.990	1.000	0.995	1.000	0.000	0.994	0.000	1.000	0.000	1.000	0.010	0.000	91000
		3	0.988	1.000	0.994	1.000	0.000	0.993	0.000	1.000	0.000	1.000	0.012	0.000	37000
30	1	1	0.185	1.000	0.312	0.950	0.050	0.414	0.051	0.949	0.000	1.000	0.815	0.000	220000
		2	0.358	1.000	0.524	0.974	0.026	0.570	0.026	0.974	0.000	1.000	0.642	0.000	156000
		3	0.528	1.000	0.686	0.984	0.016	0.694	0.016	0.984	0.000	1.000	0.472	0.000	97000
	2	1	0.839	1.000	0.911	0.996	0.004	0.903	0.004	0.996	0.000	1.000	0.161	0.000	64000
		2	0.969	1.000	0.984	0.999	0.001	0.982	0.001	0.999	0.000	1.000	0.031	0.000	43000
		3	0.978	1.000	0.989	1.000	0.000	0.987	0.000	1.000	0.000	1.000	0.022	0.000	46000
	3	1	0.979	1.000	0.989	1.000	0.000	0.987	0.000	1.000	0.000	1.000	0.021	0.000	50000
		2	0.990	1.000	0.995	1.000	0.000	0.994	0.000	1.000	0.000	1.000	0.010	0.000	62000
		3	0.975	1.000	0.987	0.999	0.001	0.985	0.001	0.999	0.000	1.000	0.025	0.000	63000
40	1	1	0.244	1.000	0.390	0.960	0.040	0.470	0.041	0.959	0.000	1.000	0.756	0.000	175000
		2	0.462	1.000	0.628	0.982	0.018	0.650	0.018	0.982	0.000	1.000	0.538	0.000	104000
		3	0.580	1.000	0.730	0.988	0.012	0.736	0.012	0.988	0.000	1.000	0.420	0.000	81000
	2	1	0.899	1.000	0.946	0.998	0.002	0.939	0.002	0.998	0.000	1.000	0.101	0.000	66000
		2	0.980	1.000	0.990	1.000	0.000	0.988	0.000	1.000	0.000	1.000	0.020	0.000	56000
		3	0.991	1.000	0.995	1.000	0.000	0.995	0.000	1.000	0.000	1.000	0.009	0.000	91000
	3	1	0.978	1.000	0.989	1.000	0.000	0.987	0.000	1.000	0.000	1.000	0.022	0.000	88000
		2	0.976	1.000	0.987	1.000	0.000	0.985	0.000	1.000	0.000	1.000	0.024	0.000	61000
		3	0.960	1.000	0.979	0.999	0.001	0.976	0.001	0.999	0.000	1.000	0.040	0.000	83000
50	1	1	0.324	1.000	0.486	0.969	0.031	0.539	0.031	0.969	0.000	1.000	0.676	0.000	173000
		2	0.565	1.000	0.717	0.986	0.014	0.721	0.014	0.986	0.000	1.000	0.435	0.000	118000
		3	0.629	1.000	0.768	0.990	0.010	0.770	0.010	0.990	0.000	1.000	0.371	0.000	90000
	2	1	0.940	1.000	0.968	0.999	0.001	0.964	0.001	0.999	0.000	1.000	0.060	0.000	70000
		2	0.985	1.000	0.992	1.000	0.000	0.991	0.000	1.000	0.000	1.000	0.015	0.000	55000
		3	0.991	1.000	0.995	1.000	0.000	0.995	0.000	1.000	0.000	1.000	0.009	0.000	56000
	3	1	0.991	1.000	0.995	1.000	0.000	0.995	0.000	1.000	0.000	1.000	0.009	0.000	66000
		2	0.987	1.000	0.993	1.000	0.000	0.992	0.000	1.000	0.000	1.000	0.013	0.000	68000
		3	0.985	1.000	0.992	1.000	0.000	0.991	0.000	1.000	0.000	1.000	0.016	0.000	58000

Table C.23: SBR all results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$F$	$\$f$	$\$c$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
10	1	1	0.060	1.000	0.113	0.682	0.318	0.201	0.325	0.675	0.000	1.000	0.940	0.000	1030129
		2	0.111	1.000	0.200	0.857	0.143	0.303	0.145	0.855	0.000	1.000	0.889	0.000	465595
		3	0.156	1.000	0.269	0.906	0.094	0.369	0.095	0.905	0.000	1.000	0.844	0.000	334137
	2	1	0.143	1.000	0.249	0.846	0.154	0.337	0.157	0.843	0.000	1.000	0.857	0.000	550904
		2	0.337	1.000	0.501	0.963	0.037	0.544	0.038	0.962	0.000	1.000	0.663	0.000	192749
		3	0.521	1.000	0.683	0.983	0.017	0.692	0.018	0.982	0.000	1.000	0.479	0.000	108482
	3	1	0.192	1.000	0.321	0.860	0.140	0.384	0.143	0.857	0.000	1.000	0.808	0.000	541813
		2	0.575	1.000	0.728	0.986	0.014	0.727	0.015	0.985	0.000	1.000	0.425	0.000	121246
		3	0.801	1.000	0.888	0.995	0.005	0.878	0.005	0.995	0.000	1.000	0.199	0.000	73137
20	1	1	0.129	1.000	0.229	0.863	0.137	0.331	0.139	0.861	0.000	1.000	0.871	0.000	487158
		2	0.228	1.000	0.370	0.931	0.069	0.452	0.071	0.929	0.000	1.000	0.772	0.000	256481
		3	0.294	1.000	0.452	0.953	0.047	0.517	0.048	0.952	0.000	1.000	0.706	0.000	178863
	2	1	0.405	1.000	0.574	0.961	0.039	0.609	0.039	0.961	0.000	1.000	0.595	0.000	194137
		2	0.722	1.000	0.837	0.991	0.009	0.830	0.009	0.991	0.000	1.000	0.278	0.000	90430
		3	0.823	1.000	0.901	0.995	0.005	0.894	0.005	0.995	0.000	1.000	0.177	0.000	88737
	3	1	0.623	1.000	0.766	0.983	0.017	0.768	0.018	0.982	0.000	1.000	0.377	0.000	151424
		2	0.901	1.000	0.947	0.998	0.002	0.941	0.002	0.998	0.000	1.000	0.099	0.000	96304
		3	0.946	1.000	0.972	0.999	0.001	0.968	0.001	0.999	0.000	1.000	0.054	0.000	73237
30	1	1	0.178	1.000	0.302	0.904	0.096	0.398	0.098	0.902	0.000	1.000	0.822	0.000	339161
		2	0.317	1.000	0.479	0.952	0.048	0.539	0.049	0.951	0.000	1.000	0.683	0.000	191395
		3	0.433	1.000	0.602	0.970	0.030	0.632	0.030	0.970	0.000	1.000	0.567	0.000	150357
	2	1	0.592	1.000	0.742	0.981	0.019	0.750	0.019	0.981	0.000	1.000	0.408	0.000	133053
		2	0.850	1.000	0.918	0.996	0.004	0.911	0.004	0.996	0.000	1.000	0.150	0.000	95094
		3	0.919	1.000	0.957	0.998	0.002	0.952	0.002	0.998	0.000	1.000	0.081	0.000	76161
	3	1	0.833	1.000	0.908	0.994	0.006	0.903	0.006	0.994	0.000	1.000	0.167	0.000	147904
		2	0.955	1.000	0.977	0.999	0.001	0.973	0.001	0.999	0.000	1.000	0.045	0.000	96213
		3	0.928	1.000	0.962	0.998	0.002	0.957	0.002	0.998	0.000	1.000	0.072	0.000	76079
40	1	1	0.233	1.000	0.377	0.929	0.071	0.459	0.072	0.928	0.000	1.000	0.767	0.000	243649
		2	0.382	1.000	0.551	0.966	0.034	0.597	0.035	0.965	0.000	1.000	0.618	0.000	155117
		3	0.493	1.000	0.658	0.977	0.023	0.681	0.024	0.976	0.000	1.000	0.507	0.000	122475
	2	1	0.722	1.000	0.837	0.989	0.011	0.835	0.011	0.989	0.000	1.000	0.278	0.000	131713
		2	0.912	1.000	0.954	0.998	0.002	0.949	0.002	0.998	0.000	1.000	0.088	0.000	97711
		3	0.959	1.000	0.979	0.999	0.001	0.976	0.001	0.999	0.000	1.000	0.041	0.000	91453
	3	1	0.894	1.000	0.944	0.997	0.003	0.940	0.003	0.997	0.000	1.000	0.106	0.000	159526
		2	0.942	1.000	0.969	0.999	0.001	0.966	0.001	0.999	0.000	1.000	0.058	0.000	101930
		3	0.916	1.000	0.955	0.998	0.002	0.950	0.002	0.998	0.000	1.000	0.084	0.000	81737
50	1	1	0.282	1.000	0.439	0.942	0.058	0.507	0.059	0.941	0.000	1.000	0.718	0.000	227231
		2	0.497	1.000	0.661	0.975	0.025	0.681	0.026	0.974	0.000	1.000	0.503	0.000	124874
		3	0.574	1.000	0.726	0.982	0.018	0.738	0.018	0.982	0.000	1.000	0.426	0.000	121114
	2	1	0.820	1.000	0.900	0.994	0.006	0.895	0.006	0.994	0.000	1.000	0.180	0.000	119281
		2	0.947	1.000	0.973	0.999	0.001	0.969	0.001	0.999	0.000	1.000	0.053	0.000	95135
		3	0.974	1.000	0.987	0.999	0.001	0.985	0.001	0.999	0.000	1.000	0.026	0.000	91526
	3	1	0.936	1.000	0.966	0.998	0.002	0.963	0.002	0.998	0.000	1.000	0.064	0.000	151687
		2	0.957	1.000	0.978	0.999	0.001	0.975	0.001	0.999	0.000	1.000	0.043	0.000	81579
		3	0.959	1.000	0.978	0.999	0.001	0.976	0.001	0.999	0.000	1.000	0.041	0.000	73249

Table C.24: SBR all results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$g=100$, $\$d=11$, $\$b=1$, $\$B=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$F$	$\$f$	$\$c$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time (ns)
10	1	1	0.061	1.000	0.116	0.540	0.460	0.179	0.474	0.526	0.000	1.000	0.939	0.000	2500439
		2	0.106	1.000	0.191	0.798	0.202	0.285	0.206	0.794	0.000	1.000	0.894	0.000	612724
		3	0.137	1.000	0.240	0.866	0.134	0.337	0.137	0.863	0.000	1.000	0.863	0.000	453216
	2	1	0.099	1.000	0.179	0.657	0.343	0.247	0.354	0.646	0.000	1.000	0.901	0.000	1400142
		2	0.226	1.000	0.367	0.926	0.074	0.433	0.075	0.925	0.000	1.000	0.774	0.000	279192
		3	0.372	1.000	0.540	0.967	0.033	0.573	0.034	0.966	0.000	1.000	0.628	0.000	140806
	3	1	0.104	1.000	0.188	0.594	0.406	0.237	0.418	0.582	0.000	1.000	0.896	0.000	1965743
		2	0.360	1.000	0.527	0.962	0.038	0.556	0.039	0.961	0.000	1.000	0.640	0.000	192750
		3	0.613	1.000	0.758	0.987	0.013	0.753	0.013	0.987	0.000	1.000	0.387	0.000	129861
20	1	1	0.129	1.000	0.229	0.801	0.199	0.319	0.205	0.795	0.000	1.000	0.871	0.000	633594
		2	0.209	1.000	0.345	0.898	0.102	0.426	0.105	0.895	0.000	1.000	0.791	0.000	357626
		3	0.274	1.000	0.428	0.932	0.068	0.497	0.070	0.930	0.000	1.000	0.726	0.000	246164
	2	1	0.294	1.000	0.453	0.913	0.087	0.507	0.090	0.910	0.000	1.000	0.706	0.000	346963
		2	0.548	1.000	0.705	0.980	0.020	0.713	0.020	0.980	0.000	1.000	0.452	0.000	134834
		3	0.704	1.000	0.825	0.990	0.010	0.819	0.010	0.990	0.000	1.000	0.296	0.000	112840
	3	1	0.419	1.000	0.590	0.942	0.058	0.611	0.060	0.940	0.000	1.000	0.581	0.000	349813
		2	0.747	1.000	0.853	0.993	0.007	0.844	0.008	0.992	0.000	1.000	0.253	0.000	175968
		3	0.867	1.000	0.928	0.997	0.003	0.920	0.003	0.997	0.000	1.000	0.133	0.000	130389
30	1	1	0.179	1.000	0.304	0.859	0.141	0.389	0.145	0.855	0.000	1.000	0.821	0.000	456578
		2	0.293	1.000	0.452	0.931	0.069	0.514	0.070	0.930	0.000	1.000	0.707	0.000	243112
		3	0.382	1.000	0.550	0.955	0.045	0.592	0.046	0.954	0.000	1.000	0.618	0.000	182568
	2	1	0.483	1.000	0.650	0.960	0.040	0.670	0.041	0.959	0.000	1.000	0.517	0.000	215013
		2	0.744	1.000	0.852	0.991	0.009	0.846	0.009	0.991	0.000	1.000	0.256	0.000	127754
		3	0.852	1.000	0.919	0.996	0.004	0.912	0.004	0.996	0.000	1.000	0.148	0.000	105377
	3	1	0.653	1.000	0.790	0.980	0.020	0.791	0.021	0.979	0.000	1.000	0.347	0.000	322521
		2	0.878	1.000	0.934	0.997	0.003	0.927	0.003	0.997	0.000	1.000	0.122	0.000	143316
		3	0.868	1.000	0.929	0.997	0.003	0.922	0.003	0.997	0.000	1.000	0.132	0.000	115527
40	1	1	0.225	1.000	0.367	0.894	0.106	0.444	0.109	0.891	0.000	1.000	0.775	0.000	305684
		2	0.378	1.000	0.547	0.952	0.048	0.592	0.049	0.951	0.000	1.000	0.622	0.000	178231
		3	0.466	1.000	0.634	0.967	0.033	0.661	0.034	0.966	0.000	1.000	0.534	0.000	154124
	2	1	0.618	1.000	0.763	0.977	0.023	0.768	0.024	0.976	0.000	1.000	0.382	0.000	196520
		2	0.844	1.000	0.914	0.995	0.005	0.908	0.005	0.995	0.000	1.000	0.156	0.000	136851
		3	0.919	1.000	0.957	0.998	0.002	0.953	0.002	0.998	0.000	1.000	0.081	0.000	123465
	3	1	0.782	1.000	0.877	0.989	0.011	0.873	0.011	0.989	0.000	1.000	0.218	0.000	307400
		2	0.897	1.000	0.945	0.997	0.003	0.939	0.003	0.997	0.000	1.000	0.103	0.000	123477
		3	0.858	1.000	0.922	0.996	0.004	0.916	0.004	0.996	0.000	1.000	0.142	0.000	109212
50	1	1	0.281	1.000	0.438	0.918	0.082	0.502	0.084	0.916	0.000	1.000	0.719	0.000	274620
		2	0.455	1.000	0.624	0.963	0.037	0.652	0.038	0.962	0.000	1.000	0.545	0.000	161793
		3	0.531	1.000	0.691	0.974	0.026	0.709	0.027	0.973	0.000	1.000	0.469	0.000	139791
	2	1	0.724	1.000	0.839	0.985	0.015	0.837	0.015	0.985	0.000	1.000	0.276	0.000	178608
		2	0.889	1.000	0.941	0.996	0.004	0.936	0.004	0.996	0.000	1.000	0.111	0.000	138155
		3	0.946	1.000	0.972	0.998	0.002	0.969	0.002	0.998	0.000	1.000	0.054	0.000	129759
	3	1	0.859	1.000	0.923	0.994	0.006	0.920	0.007	0.993	0.000	1.000	0.141	0.000	264337
		2	0.897	1.000	0.945	0.997	0.003	0.940	0.003	0.997	0.000	1.000	0.103	0.000	135416
		3	0.928	1.000	0.962	0.998	0.002	0.958	0.002	0.998	0.000	1.000	0.072	0.000	104510

Table C.25: PEC all results for generated simulations based on following parameters:
 $\mathcal{P}=1$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$d$	$\$F$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time(ns)
25	5	10	0.988	0.885	0.929	0.995	0.005	0.879	0.001	0.999	0.115	0.995	0.012	0.005	605000
		30	0.994	0.995	0.994	0.999	0.001	0.991	0.001	0.999	0.005	1.000	0.006	0.000	272500
		50	0.989	0.993	0.990	0.999	0.001	0.986	0.001	0.999	0.007	1.000	0.011	0.000	285000
	8	10	0.968	0.970	0.968	0.996	0.004	0.952	0.002	0.998	0.030	0.999	0.032	0.001	474286
		30	0.979	0.984	0.981	0.998	0.002	0.972	0.002	0.998	0.016	0.999	0.021	0.001	311429
		50	0.997	0.994	0.995	1.000	0.000	0.993	0.000	1.000	0.006	1.000	0.003	0.000	234286
	11	10	0.992	0.978	0.984	0.998	0.002	0.973	0.001	0.999	0.022	0.999	0.008	0.001	482000
		30	0.977	0.997	0.987	0.998	0.002	0.983	0.002	0.998	0.003	1.000	0.023	0.000	222000
		50	0.995	0.999	0.997	1.000	0.000	0.996	0.000	1.000	0.001	1.000	0.005	0.000	217000
50	5	10	0.993	0.880	0.924	0.997	0.003	0.876	0.000	1.000	0.120	0.998	0.007	0.002	955000
		30	0.968	0.948	0.953	0.998	0.002	0.929	0.001	0.999	0.052	0.999	0.032	0.001	467500
		50	0.986	0.970	0.976	0.999	0.001	0.962	0.001	0.999	0.030	0.999	0.014	0.001	417500
	8	10	0.959	0.907	0.929	0.997	0.003	0.885	0.002	0.998	0.093	0.998	0.041	0.002	1025714
		30	0.963	0.977	0.969	0.998	0.002	0.956	0.001	0.999	0.023	1.000	0.037	0.000	425714
		50	0.974	0.984	0.978	0.999	0.001	0.969	0.001	0.999	0.016	1.000	0.026	0.000	315714
	11	10	0.976	0.939	0.956	0.998	0.002	0.926	0.001	0.999	0.061	0.999	0.024	0.001	896000
		30	0.978	0.987	0.982	0.999	0.001	0.974	0.001	0.999	0.013	1.000	0.022	0.000	426000
		50	0.970	0.998	0.983	0.999	0.001	0.980	0.001	0.999	0.002	1.000	0.030	0.000	327000
75	5	10	0.989	0.775	0.856	0.997	0.003	0.769	0.000	1.000	0.225	0.997	0.011	0.003	1362500
		30	0.965	0.945	0.951	0.998	0.002	0.926	0.001	0.999	0.055	0.999	0.035	0.001	597500
		50	0.992	0.960	0.973	0.999	0.001	0.956	0.000	1.000	0.040	0.999	0.008	0.001	395000
	8	10	0.983	0.890	0.930	0.998	0.002	0.881	0.000	1.000	0.110	0.999	0.017	0.001	1310000
		30	0.976	0.970	0.972	0.999	0.001	0.956	0.001	0.999	0.030	1.000	0.024	0.000	617143
		50	0.972	0.986	0.978	0.999	0.001	0.970	0.001	0.999	0.014	1.000	0.028	0.000	405714
	11	10	0.976	0.926	0.949	0.998	0.002	0.914	0.001	0.999	0.074	0.999	0.024	0.001	1369000
		30	0.983	0.971	0.976	0.999	0.001	0.961	0.000	1.000	0.029	1.000	0.017	0.000	589000
		50	0.975	0.985	0.979	0.999	0.001	0.970	0.001	0.999	0.015	1.000	0.025	0.000	418000
100	5	10	0.953	0.698	0.782	0.996	0.004	0.683	0.001	0.999	0.302	0.997	0.047	0.003	2022500
		30	0.995	0.948	0.966	0.999	0.001	0.945	0.000	1.000	0.052	0.999	0.005	0.001	712500
		50	0.989	0.965	0.974	0.999	0.001	0.958	0.000	1.000	0.035	1.000	0.011	0.000	527500
	8	10	0.971	0.867	0.913	0.998	0.002	0.853	0.000	1.000	0.133	0.999	0.029	0.001	1777143
		30	0.957	0.960	0.957	0.999	0.001	0.940	0.001	0.999	0.040	1.000	0.043	0.000	737143
		50	0.987	0.984	0.985	1.000	0.000	0.977	0.000	1.000	0.016	1.000	0.013	0.000	515714
	11	10	0.982	0.930	0.954	0.999	0.001	0.921	0.000	1.000	0.070	0.999	0.018	0.001	1834000
		30	0.996	0.975	0.985	1.000	0.000	0.973	0.000	1.000	0.025	1.000	0.004	0.000	733000
		50	0.992	0.983	0.987	1.000	0.000	0.978	0.000	1.000	0.017	1.000	0.008	0.000	471000

Table C.26: PEC all results for generated simulations based on following parameters:
 $\mathcal{P}=2$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$d$	$\$F$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time(ns)
25	5	10	0.937	0.792	0.849	0.980	0.020	0.809	0.005	0.995	0.208	0.984	0.063	0.016	1107500
		30	0.993	0.960	0.975	0.996	0.004	0.970	0.001	0.999	0.040	0.997	0.007	0.003	650893
		50	0.971	0.931	0.945	0.992	0.008	0.932	0.003	0.997	0.069	0.994	0.029	0.006	547857
	8	10	0.960	0.927	0.941	0.991	0.009	0.923	0.004	0.996	0.073	0.994	0.040	0.006	975101
		30	0.971	0.959	0.960	0.994	0.006	0.950	0.003	0.997	0.041	0.997	0.029	0.003	557637
		50	0.996	0.942	0.967	0.995	0.005	0.958	0.000	1.000	0.058	0.995	0.004	0.005	647088
	11	10	0.985	0.955	0.968	0.995	0.005	0.956	0.002	0.998	0.045	0.997	0.015	0.003	847798
		30	0.980	0.958	0.965	0.995	0.005	0.957	0.002	0.998	0.042	0.997	0.020	0.003	598711
		50	0.997	0.975	0.983	0.998	0.002	0.980	0.000	1.000	0.025	0.998	0.003	0.002	438395
50	5	10	0.937	0.762	0.833	0.989	0.011	0.794	0.002	0.998	0.237	0.991	0.063	0.009	1854702
		30	0.941	0.848	0.886	0.992	0.008	0.850	0.003	0.997	0.152	0.995	0.059	0.005	986964
		50	0.987	0.908	0.944	0.996	0.004	0.925	0.001	0.999	0.092	0.996	0.013	0.004	799643
	8	10	0.904	0.798	0.843	0.989	0.011	0.801	0.004	0.996	0.202	0.993	0.096	0.007	1886868
		30	0.955	0.921	0.934	0.995	0.005	0.917	0.002	0.998	0.079	0.997	0.045	0.003	862179
		50	0.959	0.937	0.945	0.995	0.005	0.932	0.002	0.998	0.063	0.998	0.041	0.002	704176
	11	10	0.916	0.872	0.891	0.992	0.008	0.859	0.004	0.996	0.128	0.995	0.084	0.005	1704158
		30	0.968	0.943	0.953	0.996	0.004	0.941	0.002	0.998	0.057	0.998	0.032	0.002	854249
		50	0.960	0.975	0.967	0.997	0.003	0.960	0.002	0.998	0.025	0.999	0.040	0.001	710208
75	5	10	0.899	0.659	0.751	0.990	0.010	0.689	0.002	0.998	0.341	0.992	0.101	0.008	2570655
		30	0.950	0.827	0.878	0.994	0.006	0.845	0.001	0.999	0.173	0.996	0.050	0.004	1186429
		50	0.986	0.858	0.914	0.996	0.004	0.879	0.000	1.000	0.142	0.997	0.014	0.003	945179
	8	10	0.882	0.778	0.822	0.992	0.008	0.772	0.003	0.997	0.222	0.995	0.118	0.005	2478306
		30	0.955	0.892	0.919	0.996	0.004	0.892	0.001	0.999	0.108	0.997	0.045	0.003	1131832
		50	0.971	0.934	0.951	0.997	0.003	0.936	0.001	0.999	0.066	0.998	0.029	0.002	906978
	11	10	0.930	0.837	0.877	0.994	0.006	0.842	0.002	0.998	0.163	0.996	0.070	0.004	2330854
		30	0.977	0.921	0.947	0.997	0.003	0.929	0.001	0.999	0.079	0.998	0.023	0.002	1125268
		50	0.961	0.951	0.953	0.997	0.003	0.943	0.001	0.999	0.049	0.999	0.039	0.001	817830
100	5	10	0.845	0.551	0.657	0.991	0.009	0.575	0.002	0.998	0.449	0.992	0.155	0.008	3514107
		30	0.980	0.812	0.883	0.996	0.004	0.842	0.000	1.000	0.188	0.996	0.020	0.004	1445000
		50	0.980	0.864	0.916	0.997	0.003	0.889	0.000	1.000	0.136	0.997	0.020	0.003	1030179
	8	10	0.896	0.721	0.794	0.994	0.006	0.733	0.002	0.998	0.279	0.995	0.104	0.005	3302747
		30	0.949	0.888	0.914	0.997	0.003	0.889	0.001	0.999	0.112	0.998	0.051	0.002	1344725
		50	0.977	0.922	0.948	0.998	0.002	0.929	0.001	0.999	0.078	0.999	0.023	0.001	1043727
	11	10	0.938	0.818	0.869	0.995	0.005	0.830	0.001	0.999	0.182	0.997	0.062	0.003	3031313
		30	0.981	0.926	0.952	0.998	0.002	0.934	0.000	1.000	0.074	0.999	0.019	0.001	1315395
		50	0.987	0.929	0.954	0.998	0.002	0.939	0.000	1.000	0.071	0.999	0.013	0.001	933026

Table C.27: PEC all results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$d$	$\$F$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time(ns)
25	5	10	0.910	0.726	0.799	0.966	0.034	0.761	0.008	0.992	0.274	0.971	0.090	0.029	1323732
		30	0.997	0.944	0.968	0.994	0.006	0.964	0.000	1.000	0.056	0.994	0.003	0.006	698328
		50	0.987	0.921	0.949	0.990	0.010	0.938	0.002	0.998	0.079	0.991	0.013	0.009	757045
	8	10	0.960	0.874	0.912	0.983	0.017	0.893	0.005	0.995	0.126	0.986	0.040	0.014	1142979
		30	0.967	0.930	0.938	0.991	0.009	0.925	0.003	0.997	0.070	0.994	0.033	0.006	615487
		50	0.996	0.914	0.950	0.990	0.010	0.940	0.001	0.999	0.086	0.991	0.004	0.009	730897
	11	10	0.996	0.930	0.960	0.994	0.006	0.944	0.000	1.000	0.070	0.994	0.004	0.006	840305
		30	0.987	0.943	0.962	0.992	0.008	0.954	0.002	0.998	0.057	0.994	0.013	0.006	779858
		50	1.000	0.951	0.969	0.995	0.005	0.966	0.000	1.000	0.049	0.994	0.000	0.006	509839
50	5	10	0.899	0.682	0.769	0.981	0.019	0.726	0.004	0.996	0.318	0.984	0.101	0.016	2320626
		30	0.948	0.833	0.883	0.989	0.011	0.853	0.003	0.997	0.166	0.992	0.052	0.008	1233273
		50	0.985	0.833	0.897	0.990	0.010	0.879	0.001	0.999	0.167	0.991	0.015	0.009	1012788
	8	10	0.835	0.680	0.744	0.978	0.022	0.696	0.007	0.993	0.320	0.984	0.165	0.016	2415905
		30	0.960	0.903	0.927	0.992	0.008	0.913	0.003	0.997	0.097	0.995	0.040	0.005	1143031
		50	0.959	0.933	0.944	0.994	0.006	0.932	0.003	0.997	0.067	0.997	0.041	0.003	929375
	11	10	0.871	0.804	0.832	0.984	0.016	0.800	0.007	0.993	0.196	0.990	0.129	0.010	2246629
		30	0.963	0.925	0.942	0.994	0.006	0.932	0.003	0.997	0.075	0.996	0.037	0.004	1107689
		50	0.950	0.952	0.949	0.994	0.006	0.942	0.004	0.996	0.048	0.997	0.050	0.003	931262
75	5	10	0.839	0.548	0.652	0.982	0.018	0.593	0.004	0.996	0.452	0.985	0.161	0.015	3182197
		30	0.943	0.787	0.852	0.991	0.009	0.815	0.002	0.998	0.213	0.993	0.057	0.007	1520985
		50	0.982	0.771	0.857	0.992	0.008	0.817	0.001	0.999	0.229	0.993	0.018	0.007	1150227
	8	10	0.842	0.673	0.740	0.985	0.015	0.696	0.005	0.995	0.327	0.989	0.158	0.011	3212126
		30	0.951	0.858	0.899	0.993	0.007	0.876	0.002	0.998	0.142	0.995	0.049	0.005	1460459
		50	0.977	0.902	0.936	0.996	0.004	0.920	0.001	0.999	0.098	0.997	0.023	0.003	1114648
	11	10	0.874	0.754	0.807	0.989	0.011	0.774	0.004	0.996	0.246	0.992	0.126	0.008	3168019
		30	0.973	0.918	0.944	0.996	0.004	0.929	0.001	0.999	0.082	0.997	0.027	0.003	1441067
		50	0.970	0.953	0.961	0.997	0.003	0.954	0.001	0.999	0.047	0.998	0.030	0.002	1085123
100	5	10	0.807	0.452	0.567	0.984	0.016	0.503	0.003	0.997	0.548	0.987	0.193	0.013	4515025
		30	0.961	0.774	0.853	0.994	0.006	0.818	0.001	0.999	0.226	0.994	0.039	0.006	1823182
		50	0.978	0.806	0.877	0.994	0.006	0.850	0.001	0.999	0.194	0.995	0.022	0.005	1349470
	8	10	0.834	0.630	0.711	0.988	0.012	0.662	0.003	0.997	0.370	0.991	0.166	0.009	4387130
		30	0.949	0.867	0.903	0.995	0.005	0.883	0.002	0.998	0.133	0.997	0.051	0.003	1797639
		50	0.979	0.885	0.928	0.996	0.004	0.907	0.001	0.999	0.115	0.997	0.021	0.003	1335437
	11	10	0.897	0.754	0.815	0.992	0.008	0.781	0.002	0.998	0.246	0.994	0.103	0.006	4115618
		30	0.972	0.883	0.923	0.996	0.004	0.905	0.001	0.999	0.117	0.997	0.028	0.003	1761252
		50	0.985	0.899	0.935	0.997	0.003	0.922	0.001	0.999	0.101	0.997	0.015	0.003	1252194

Table C.28: SBR all results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$d$	$\$F$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time(ns)
25	5	10	0.456	1.000	0.618	0.928	0.072	0.633	0.075	0.925	0.000	1.000	0.544	0.000	40000
		30	0.774	1.000	0.866	0.977	0.023	0.855	0.024	0.976	0.000	1.000	0.226	0.000	17500
		50	0.857	1.000	0.920	0.988	0.012	0.911	0.013	0.987	0.000	1.000	0.143	0.000	47500
	8	10	0.498	1.000	0.658	0.934	0.066	0.663	0.068	0.932	0.000	1.000	0.502	0.000	38571
		30	0.730	1.000	0.841	0.974	0.026	0.828	0.027	0.973	0.000	1.000	0.270	0.000	31429
		50	0.832	1.000	0.905	0.986	0.014	0.895	0.015	0.985	0.000	1.000	0.168	0.000	50000
	11	10	0.497	1.000	0.658	0.935	0.065	0.660	0.068	0.932	0.000	1.000	0.503	0.000	77000
		30	0.762	1.000	0.862	0.978	0.022	0.849	0.023	0.977	0.000	1.000	0.238	0.000	43000
		50	0.851	1.000	0.918	0.987	0.013	0.907	0.013	0.987	0.000	1.000	0.149	0.000	45000
50	5	10	0.246	1.000	0.393	0.926	0.074	0.470	0.075	0.925	0.000	1.000	0.754	0.000	50000
		30	0.586	1.000	0.730	0.976	0.024	0.736	0.024	0.976	0.000	1.000	0.414	0.000	42500
		50	0.694	1.000	0.810	0.983	0.017	0.806	0.017	0.983	0.000	1.000	0.306	0.000	42500
	8	10	0.216	1.000	0.355	0.917	0.083	0.440	0.085	0.915	0.000	1.000	0.784	0.000	94286
		30	0.582	1.000	0.730	0.976	0.024	0.733	0.025	0.975	0.000	1.000	0.418	0.000	40000
		50	0.708	1.000	0.824	0.985	0.015	0.816	0.016	0.984	0.000	1.000	0.292	0.000	45714
	11	10	0.247	1.000	0.395	0.925	0.075	0.469	0.077	0.923	0.000	1.000	0.753	0.000	154000
		30	0.584	1.000	0.733	0.975	0.025	0.732	0.025	0.975	0.000	1.000	0.416	0.000	87000
		50	0.673	1.000	0.801	0.983	0.017	0.795	0.017	0.983	0.000	1.000	0.327	0.000	73000
75	5	10	0.171	1.000	0.291	0.923	0.077	0.390	0.078	0.922	0.000	1.000	0.829	0.000	80000
		30	0.419	1.000	0.583	0.974	0.026	0.622	0.027	0.973	0.000	1.000	0.581	0.000	62500
		50	0.614	1.000	0.750	0.986	0.014	0.757	0.015	0.985	0.000	1.000	0.386	0.000	47500
	8	10	0.169	1.000	0.288	0.925	0.075	0.390	0.076	0.924	0.000	1.000	0.831	0.000	102857
		30	0.434	1.000	0.599	0.973	0.027	0.628	0.027	0.973	0.000	1.000	0.566	0.000	50000
		50	0.578	1.000	0.725	0.984	0.016	0.733	0.016	0.984	0.000	1.000	0.422	0.000	52857
	11	10	0.164	1.000	0.281	0.925	0.075	0.385	0.076	0.924	0.000	1.000	0.836	0.000	221000
		30	0.475	1.000	0.639	0.975	0.025	0.653	0.025	0.974	0.000	1.000	0.525	0.000	99000
		50	0.592	1.000	0.740	0.985	0.015	0.743	0.015	0.985	0.000	1.000	0.408	0.000	105000
100	5	10	0.113	1.000	0.203	0.912	0.088	0.318	0.088	0.912	0.000	1.000	0.887	0.000	67500
		30	0.346	1.000	0.507	0.974	0.026	0.563	0.026	0.974	0.000	1.000	0.654	0.000	50000
		50	0.523	1.000	0.677	0.984	0.016	0.692	0.016	0.984	0.000	1.000	0.477	0.000	60000
	8	10	0.120	1.000	0.214	0.921	0.079	0.330	0.080	0.920	0.000	1.000	0.880	0.000	185714
		30	0.343	1.000	0.506	0.974	0.026	0.561	0.027	0.973	0.000	1.000	0.657	0.000	82857
		50	0.502	1.000	0.663	0.984	0.016	0.679	0.016	0.984	0.000	1.000	0.498	0.000	62857
	11	10	0.127	1.000	0.225	0.921	0.079	0.337	0.079	0.921	0.000	1.000	0.873	0.000	329000
		30	0.358	1.000	0.524	0.974	0.026	0.570	0.026	0.974	0.000	1.000	0.642	0.000	156000
		50	0.565	1.000	0.717	0.986	0.014	0.721	0.014	0.986	0.000	1.000	0.435	0.000	118000

Table C.29: SBR all results for generated simulations based on following parameters: $\mathcal{P}=2$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$d$	$\$F$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time(ns)
25	5	10	0.451	1.000	0.616	0.892	0.108	0.621	0.117	0.883	0.000	1.000	0.549	0.000	42143
		30	0.783	1.000	0.873	0.968	0.032	0.861	0.034	0.966	0.000	1.000	0.217	0.000	33750
		50	0.845	1.000	0.913	0.980	0.020	0.904	0.021	0.979	0.000	1.000	0.155	0.000	51250
	8	10	0.451	1.000	0.617	0.888	0.112	0.617	0.120	0.880	0.000	1.000	0.549	0.000	50559
		30	0.712	1.000	0.827	0.964	0.036	0.816	0.039	0.961	0.000	1.000	0.288	0.000	47308
		50	0.783	1.000	0.876	0.972	0.028	0.865	0.030	0.970	0.000	1.000	0.217	0.000	45604
	11	10	0.455	1.000	0.620	0.903	0.097	0.622	0.103	0.897	0.000	1.000	0.545	0.000	92044
		30	0.696	1.000	0.819	0.956	0.044	0.805	0.047	0.953	0.000	1.000	0.304	0.000	56842
		50	0.834	1.000	0.908	0.981	0.019	0.897	0.021	0.979	0.000	1.000	0.166	0.000	45184
50	5	10	0.230	1.000	0.372	0.866	0.134	0.440	0.139	0.861	0.000	1.000	0.770	0.000	79524
		30	0.518	1.000	0.676	0.953	0.047	0.687	0.049	0.951	0.000	1.000	0.482	0.000	58036
		50	0.647	1.000	0.781	0.971	0.029	0.778	0.031	0.969	0.000	1.000	0.353	0.000	75536
	8	10	0.211	1.000	0.347	0.856	0.144	0.419	0.149	0.851	0.000	1.000	0.789	0.000	119560
		30	0.519	1.000	0.679	0.956	0.044	0.691	0.045	0.955	0.000	1.000	0.481	0.000	75229
		50	0.652	1.000	0.785	0.973	0.027	0.784	0.028	0.972	0.000	1.000	0.348	0.000	72756
	11	10	0.212	1.000	0.349	0.864	0.136	0.420	0.140	0.860	0.000	1.000	0.788	0.000	239079
		30	0.504	1.000	0.666	0.954	0.046	0.679	0.048	0.952	0.000	1.000	0.496	0.000	111719
		50	0.603	1.000	0.750	0.969	0.031	0.752	0.032	0.968	0.000	1.000	0.397	0.000	89661
75	5	10	0.160	1.000	0.274	0.869	0.131	0.368	0.134	0.866	0.000	1.000	0.840	0.000	125060
		30	0.403	1.000	0.570	0.957	0.043	0.610	0.044	0.956	0.000	1.000	0.597	0.000	58750
		50	0.556	1.000	0.711	0.975	0.025	0.724	0.026	0.974	0.000	1.000	0.444	0.000	71786
	8	10	0.152	1.000	0.264	0.863	0.137	0.357	0.140	0.860	0.000	1.000	0.848	0.000	192436
		30	0.370	1.000	0.537	0.951	0.049	0.582	0.050	0.950	0.000	1.000	0.630	0.000	94460
		50	0.514	1.000	0.675	0.970	0.030	0.694	0.030	0.970	0.000	1.000	0.486	0.000	81374
	11	10	0.149	1.000	0.258	0.862	0.138	0.352	0.141	0.859	0.000	1.000	0.851	0.000	374813
		30	0.395	1.000	0.564	0.952	0.048	0.599	0.049	0.951	0.000	1.000	0.605	0.000	163771
		50	0.543	1.000	0.701	0.973	0.027	0.714	0.028	0.972	0.000	1.000	0.457	0.000	121699
100	5	10	0.110	1.000	0.198	0.850	0.150	0.302	0.152	0.848	0.000	1.000	0.890	0.000	156964
		30	0.332	1.000	0.495	0.956	0.044	0.554	0.045	0.955	0.000	1.000	0.668	0.000	81250
		50	0.494	1.000	0.654	0.974	0.026	0.677	0.026	0.974	0.000	1.000	0.506	0.000	91786
	8	10	0.109	1.000	0.197	0.853	0.147	0.302	0.150	0.850	0.000	1.000	0.891	0.000	280604
		30	0.320	1.000	0.482	0.953	0.047	0.541	0.048	0.952	0.000	1.000	0.680	0.000	118462
		50	0.449	1.000	0.615	0.971	0.029	0.646	0.030	0.970	0.000	1.000	0.551	0.000	107967
	11	10	0.111	1.000	0.200	0.857	0.143	0.303	0.145	0.855	0.000	1.000	0.889	0.000	465595
		30	0.317	1.000	0.479	0.952	0.048	0.539	0.049	0.951	0.000	1.000	0.683	0.000	191395
		50	0.497	1.000	0.661	0.975	0.025	0.681	0.026	0.974	0.000	1.000	0.503	0.000	124874

Table C.30: SBR all results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$b=1$, $\$B=2$, $\$f=1$, $\$c=2$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$d$	$\$F$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time(ns)
25	5	10	0.432	1.000	0.599	0.855	0.145	0.596	0.161	0.839	0.000	1.000	0.568	0.000	65394
		30	0.785	1.000	0.876	0.963	0.037	0.861	0.041	0.959	0.000	1.000	0.215	0.000	46242
		50	0.858	1.000	0.921	0.977	0.023	0.910	0.026	0.974	0.000	1.000	0.142	0.000	58485
	8	10	0.439	1.000	0.607	0.855	0.145	0.598	0.160	0.840	0.000	1.000	0.561	0.000	74722
		30	0.712	1.000	0.828	0.957	0.043	0.813	0.047	0.953	0.000	1.000	0.288	0.000	48830
		50	0.744	1.000	0.852	0.959	0.041	0.838	0.046	0.954	0.000	1.000	0.256	0.000	51570
	11	10	0.455	1.000	0.619	0.897	0.103	0.621	0.111	0.889	0.000	1.000	0.545	0.000	106635
		30	0.683	1.000	0.810	0.942	0.058	0.793	0.065	0.935	0.000	1.000	0.317	0.000	77517
		50	0.805	1.000	0.890	0.974	0.026	0.877	0.028	0.972	0.000	1.000	0.195	0.000	64980
50	5	10	0.216	1.000	0.354	0.809	0.191	0.412	0.201	0.799	0.000	1.000	0.784	0.000	124354
		30	0.502	1.000	0.665	0.935	0.065	0.673	0.069	0.931	0.000	1.000	0.498	0.000	77197
		50	0.607	1.000	0.751	0.956	0.044	0.752	0.047	0.953	0.000	1.000	0.393	0.000	86727
	8	10	0.205	1.000	0.339	0.802	0.198	0.398	0.208	0.792	0.000	1.000	0.795	0.000	179900
		30	0.488	1.000	0.652	0.934	0.066	0.664	0.069	0.931	0.000	1.000	0.512	0.000	81495
		50	0.628	1.000	0.768	0.962	0.038	0.767	0.041	0.959	0.000	1.000	0.372	0.000	81041
	11	10	0.204	1.000	0.338	0.810	0.190	0.398	0.199	0.801	0.000	1.000	0.796	0.000	310522
		30	0.488	1.000	0.654	0.936	0.064	0.666	0.068	0.932	0.000	1.000	0.512	0.000	141931
		50	0.591	1.000	0.741	0.956	0.044	0.744	0.046	0.954	0.000	1.000	0.409	0.000	107617
75	5	10	0.163	1.000	0.280	0.821	0.179	0.361	0.185	0.815	0.000	1.000	0.837	0.000	152955
		30	0.380	1.000	0.547	0.938	0.062	0.589	0.065	0.935	0.000	1.000	0.620	0.000	75303
		50	0.528	1.000	0.688	0.963	0.037	0.704	0.039	0.961	0.000	1.000	0.472	0.000	86000
	8	10	0.149	1.000	0.260	0.807	0.193	0.342	0.199	0.801	0.000	1.000	0.851	0.000	232316
		30	0.357	1.000	0.524	0.931	0.069	0.568	0.072	0.928	0.000	1.000	0.643	0.000	134708
		50	0.503	1.000	0.666	0.959	0.041	0.685	0.043	0.957	0.000	1.000	0.497	0.000	100739
	11	10	0.138	1.000	0.243	0.804	0.196	0.328	0.202	0.798	0.000	1.000	0.862	0.000	475909
		30	0.379	1.000	0.548	0.934	0.066	0.585	0.068	0.932	0.000	1.000	0.621	0.000	207710
		50	0.515	1.000	0.678	0.961	0.039	0.694	0.040	0.960	0.000	1.000	0.485	0.000	150975
100	5	10	0.113	1.000	0.203	0.793	0.207	0.296	0.213	0.787	0.000	1.000	0.887	0.000	179056
		30	0.317	1.000	0.479	0.938	0.062	0.538	0.063	0.937	0.000	1.000	0.683	0.000	96995
		50	0.466	1.000	0.631	0.962	0.038	0.658	0.039	0.961	0.000	1.000	0.534	0.000	107076
	8	10	0.107	1.000	0.193	0.793	0.207	0.287	0.213	0.787	0.000	1.000	0.893	0.000	367814
		30	0.309	1.000	0.470	0.932	0.068	0.528	0.069	0.931	0.000	1.000	0.691	0.000	147138
		50	0.417	1.000	0.586	0.957	0.043	0.623	0.044	0.956	0.000	1.000	0.583	0.000	120311
	11	10	0.106	1.000	0.191	0.798	0.202	0.285	0.206	0.794	0.000	1.000	0.894	0.000	612724
		30	0.293	1.000	0.452	0.931	0.069	0.514	0.070	0.930	0.000	1.000	0.707	0.000	243112
		50	0.455	1.000	0.624	0.963	0.037	0.652	0.038	0.962	0.000	1.000	0.545	0.000	161793

Table C.31: PEC all results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$f$	$\$c$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time(ns)
25	1	1	0.995	0.969	0.981	0.998	0.002	0.966	0.000	1.000	0.031	0.999	0.005	0.001	1069000
		2	0.992	0.978	0.984	0.998	0.002	0.973	0.001	0.999	0.022	0.999	0.008	0.001	482000
		3	0.963	0.970	0.965	0.996	0.004	0.948	0.003	0.997	0.030	0.999	0.037	0.001	421000
	2	1	0.987	0.989	0.988	0.999	0.001	0.981	0.001	0.999	0.011	1.000	0.013	0.000	304000
		2	0.998	1.000	0.999	1.000	0.000	0.998	0.000	1.000	0.000	1.000	0.003	0.000	153000
		3	0.998	1.000	0.999	1.000	0.000	0.999	0.000	1.000	0.000	1.000	0.002	0.000	151000
	3	1	0.977	0.994	0.985	0.998	0.002	0.980	0.002	0.998	0.006	1.000	0.023	0.000	205000
		2	0.997	1.000	0.998	1.000	0.000	0.998	0.000	1.000	0.000	1.000	0.003	0.000	141000
		3	1.000	0.996	0.998	1.000	0.000	0.996	0.000	1.000	0.004	1.000	0.000	0.000	113000
50	1	1	0.945	0.870	0.903	0.995	0.005	0.842	0.002	0.998	0.130	0.997	0.055	0.003	2110000
		2	0.976	0.939	0.956	0.998	0.002	0.926	0.001	0.999	0.061	0.999	0.024	0.001	896000
		3	0.982	0.989	0.985	0.999	0.001	0.978	0.001	0.999	0.011	1.000	0.018	0.000	687000
	2	1	0.975	0.983	0.978	0.999	0.001	0.968	0.001	0.999	0.017	1.000	0.025	0.000	535000
		2	0.979	0.996	0.987	0.999	0.001	0.983	0.001	0.999	0.004	1.000	0.022	0.000	264000
		3	0.993	1.000	0.996	1.000	0.000	0.996	0.000	1.000	0.000	1.000	0.007	0.000	167000
	3	1	0.976	0.989	0.982	0.999	0.001	0.975	0.001	0.999	0.011	1.000	0.024	0.000	280000
		2	0.998	1.000	0.999	1.000	0.000	0.998	0.000	1.000	0.000	1.000	0.003	0.000	179000
		3	0.999	1.000	0.999	1.000	0.000	0.999	0.000	1.000	0.000	1.000	0.001	0.000	180000
75	1	1	0.943	0.858	0.896	0.997	0.003	0.830	0.001	0.999	0.142	0.998	0.057	0.002	2934000
		2	0.976	0.926	0.949	0.998	0.002	0.914	0.001	0.999	0.074	0.999	0.024	0.001	1369000
		3	0.994	0.980	0.987	1.000	0.000	0.977	0.000	1.000	0.020	1.000	0.006	0.000	947000
	2	1	0.981	0.964	0.971	0.999	0.001	0.953	0.000	1.000	0.036	1.000	0.019	0.000	638000
		2	0.990	0.990	0.990	1.000	0.000	0.984	0.000	1.000	0.010	1.000	0.010	0.000	303000
		3	0.992	0.996	0.994	1.000	0.000	0.991	0.000	1.000	0.004	1.000	0.008	0.000	203000
	3	1	0.986	0.991	0.988	1.000	0.000	0.983	0.000	1.000	0.009	1.000	0.014	0.000	352000
		2	0.996	0.998	0.997	1.000	0.000	0.995	0.000	1.000	0.002	1.000	0.004	0.000	170000
		3	0.999	1.000	0.999	1.000	0.000	0.999	0.000	1.000	0.000	1.000	0.001	0.000	163000
100	1	1	0.976	0.847	0.904	0.998	0.002	0.836	0.000	1.000	0.153	0.998	0.024	0.002	3591000
		2	0.982	0.930	0.954	0.999	0.001	0.921	0.000	1.000	0.070	0.999	0.018	0.001	1834000
		3	0.986	0.940	0.961	0.999	0.001	0.932	0.000	1.000	0.060	0.999	0.014	0.001	1292000
	2	1	0.979	0.966	0.971	0.999	0.001	0.954	0.000	1.000	0.034	1.000	0.021	0.000	915000
		2	0.969	0.978	0.973	0.999	0.001	0.961	0.001	0.999	0.022	1.000	0.031	0.000	361000
		3	0.983	0.998	0.990	1.000	0.000	0.988	0.000	1.000	0.002	1.000	0.017	0.000	232000
	3	1	0.979	0.972	0.975	0.999	0.001	0.960	0.000	1.000	0.028	1.000	0.021	0.000	464000
		2	0.993	0.999	0.996	1.000	0.000	0.995	0.000	1.000	0.001	1.000	0.008	0.000	180000
		3	0.999	0.999	0.999	1.000	0.000	0.998	0.000	1.000	0.001	1.000	0.002	0.000	178000

Table C.32: PEC all results for generated simulations based on following parameters:
 $\mathcal{P}=2$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$f$	$\$c$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time(ns)
25	1	1	0.968	0.898	0.927	0.990	0.010	0.906	0.003	0.997	0.102	0.992	0.032	0.008	1630281
		2	0.985	0.955	0.968	0.995	0.005	0.956	0.002	0.998	0.045	0.997	0.015	0.003	847798
		3	0.931	0.929	0.928	0.988	0.012	0.908	0.007	0.993	0.071	0.995	0.069	0.005	795936
	2	1	0.993	0.976	0.984	0.997	0.003	0.979	0.001	0.999	0.024	0.998	0.007	0.002	736058
		2	0.997	0.987	0.992	0.999	0.001	0.990	0.000	1.000	0.013	0.999	0.003	0.001	370871
		3	1.000	0.992	0.996	0.999	0.001	0.995	0.000	1.000	0.008	0.999	0.000	0.001	281947
	3	1	0.973	0.957	0.963	0.993	0.007	0.954	0.003	0.997	0.043	0.996	0.027	0.004	985000
		2	0.984	0.995	0.989	0.998	0.002	0.987	0.001	0.999	0.005	1.000	0.016	0.000	297363
		3	0.997	0.992	0.994	0.999	0.001	0.993	0.000	1.000	0.008	0.999	0.003	0.001	237816
50	1	1	0.817	0.695	0.741	0.982	0.018	0.691	0.007	0.993	0.305	0.988	0.183	0.012	3614234
		2	0.916	0.872	0.891	0.992	0.008	0.859	0.004	0.996	0.128	0.995	0.084	0.005	1704158
		3	0.973	0.917	0.943	0.996	0.004	0.927	0.001	0.999	0.083	0.997	0.027	0.003	1283640
	2	1	0.941	0.894	0.914	0.993	0.007	0.891	0.003	0.997	0.106	0.996	0.059	0.004	1881447
		2	0.972	0.971	0.971	0.998	0.002	0.963	0.001	0.999	0.029	0.999	0.028	0.001	618608
		3	0.987	0.986	0.986	0.999	0.001	0.984	0.001	0.999	0.014	0.999	0.013	0.001	386190
	3	1	0.977	0.946	0.960	0.997	0.003	0.950	0.001	0.999	0.054	0.998	0.023	0.002	1386956
		2	0.985	0.983	0.984	0.999	0.001	0.980	0.001	0.999	0.017	0.999	0.015	0.001	391035
		3	0.994	0.997	0.995	1.000	0.000	0.995	0.000	1.000	0.003	1.000	0.006	0.000	253289
75	1	1	0.799	0.661	0.717	0.987	0.013	0.661	0.005	0.995	0.339	0.992	0.201	0.008	5317988
		2	0.930	0.837	0.877	0.994	0.006	0.842	0.002	0.998	0.163	0.996	0.070	0.004	2330854
		3	0.980	0.898	0.935	0.997	0.003	0.915	0.001	0.999	0.102	0.997	0.020	0.003	1712962
	2	1	0.940	0.898	0.915	0.995	0.005	0.894	0.002	0.998	0.102	0.997	0.060	0.003	2404233
		2	0.982	0.942	0.961	0.998	0.002	0.949	0.001	0.999	0.058	0.999	0.018	0.001	883123
		3	0.973	0.979	0.976	0.999	0.001	0.970	0.001	0.999	0.021	0.999	0.027	0.001	491412
	3	1	0.978	0.923	0.950	0.997	0.003	0.935	0.001	0.999	0.077	0.998	0.022	0.002	2415842
		2	0.982	0.978	0.980	0.999	0.001	0.976	0.000	1.000	0.022	0.999	0.018	0.001	496763
		3	0.993	0.992	0.993	1.000	0.000	0.991	0.000	1.000	0.008	1.000	0.007	0.000	305689
100	1	1	0.795	0.618	0.689	0.990	0.010	0.639	0.003	0.997	0.382	0.993	0.205	0.007	6799351
		2	0.938	0.818	0.869	0.995	0.005	0.830	0.001	0.999	0.182	0.997	0.062	0.003	3031313
		3	0.934	0.865	0.894	0.996	0.004	0.863	0.001	0.999	0.135	0.998	0.066	0.002	2135500
	2	1	0.945	0.881	0.910	0.996	0.004	0.885	0.001	0.999	0.119	0.998	0.055	0.002	3386237
		2	0.961	0.945	0.953	0.998	0.002	0.938	0.001	0.999	0.055	0.999	0.039	0.001	1085567
		3	0.974	0.959	0.966	0.999	0.001	0.955	0.001	0.999	0.041	0.999	0.026	0.001	696009
	3	1	0.952	0.910	0.930	0.997	0.003	0.907	0.001	0.999	0.090	0.998	0.048	0.002	3044017
		2	0.972	0.977	0.974	0.999	0.001	0.968	0.001	0.999	0.023	1.000	0.028	0.000	585360
		3	0.991	0.989	0.990	1.000	0.000	0.987	0.000	1.000	0.011	1.000	0.009	0.000	374304

Table C.33: PEC all results for generated simulations based on following parameters:
 $\mathcal{P}=3$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$f$	$\$c$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time(ns)
25	1	1	0.958	0.869	0.908	0.986	0.014	0.887	0.003	0.997	0.131	0.988	0.042	0.012	1810098
		2	0.996	0.930	0.960	0.994	0.006	0.944	0.000	1.000	0.070	0.994	0.004	0.006	840305
		3	0.934	0.903	0.916	0.984	0.016	0.894	0.008	0.992	0.097	0.991	0.066	0.009	931409
	2	1	0.988	0.960	0.974	0.995	0.005	0.966	0.002	0.998	0.040	0.996	0.012	0.004	1026496
		2	0.999	0.970	0.984	0.997	0.003	0.981	0.000	1.000	0.030	0.997	0.001	0.003	415745
		3	1.000	0.986	0.993	0.999	0.001	0.991	0.000	1.000	0.014	0.999	0.000	0.001	319222
	3	1	0.943	0.906	0.922	0.982	0.018	0.908	0.009	0.991	0.094	0.989	0.057	0.011	1722839
		2	0.979	0.980	0.979	0.997	0.003	0.975	0.002	0.998	0.020	0.998	0.021	0.002	367649
		3	0.994	0.981	0.988	0.998	0.002	0.985	0.000	1.000	0.019	0.998	0.006	0.002	278117
50	1	1	0.688	0.493	0.566	0.960	0.040	0.518	0.013	0.987	0.507	0.971	0.312	0.029	5317516
		2	0.871	0.804	0.832	0.984	0.016	0.800	0.007	0.993	0.196	0.990	0.129	0.010	2246629
		3	0.957	0.854	0.901	0.991	0.009	0.880	0.002	0.998	0.146	0.993	0.043	0.007	1591080
	2	1	0.890	0.795	0.835	0.982	0.018	0.809	0.007	0.993	0.205	0.988	0.110	0.012	3496524
		2	0.966	0.947	0.956	0.996	0.004	0.947	0.002	0.998	0.053	0.998	0.034	0.002	776142
		3	0.991	0.974	0.982	0.998	0.002	0.978	0.000	1.000	0.026	0.999	0.009	0.001	484009
	3	1	0.933	0.879	0.900	0.989	0.011	0.887	0.004	0.996	0.121	0.993	0.067	0.007	2753750
		2	0.972	0.950	0.960	0.997	0.003	0.951	0.001	0.999	0.050	0.998	0.028	0.002	599954
		3	0.986	0.987	0.986	0.999	0.001	0.983	0.001	0.999	0.013	0.999	0.014	0.001	302731
75	1	1	0.734	0.500	0.585	0.975	0.025	0.538	0.007	0.993	0.500	0.981	0.266	0.019	7957823
		2	0.874	0.754	0.807	0.989	0.011	0.774	0.004	0.996	0.246	0.992	0.126	0.008	3168019
		3	0.967	0.836	0.894	0.994	0.006	0.872	0.001	0.999	0.164	0.995	0.033	0.005	2139999
	2	1	0.863	0.797	0.825	0.987	0.013	0.799	0.006	0.994	0.203	0.992	0.137	0.008	5134709
		2	0.962	0.879	0.916	0.995	0.005	0.897	0.001	0.999	0.121	0.996	0.038	0.004	1348029
		3	0.970	0.943	0.956	0.998	0.002	0.945	0.001	0.999	0.057	0.998	0.030	0.002	687975
	3	1	0.905	0.788	0.839	0.988	0.012	0.820	0.004	0.996	0.212	0.992	0.095	0.008	6495734
		2	0.967	0.946	0.955	0.998	0.002	0.947	0.001	0.999	0.054	0.999	0.033	0.001	794895
		3	0.986	0.982	0.984	0.999	0.001	0.980	0.000	1.000	0.018	1.000	0.014	0.000	366663
100	1	1	0.732	0.436	0.539	0.980	0.020	0.495	0.004	0.996	0.564	0.984	0.268	0.016	10363800
		2	0.897	0.754	0.815	0.992	0.008	0.781	0.002	0.998	0.246	0.994	0.103	0.006	4115618
		3	0.909	0.817	0.858	0.994	0.006	0.826	0.002	0.998	0.183	0.996	0.091	0.004	2876787
	2	1	0.832	0.729	0.772	0.988	0.012	0.739	0.005	0.995	0.271	0.992	0.168	0.008	7439297
		2	0.938	0.892	0.913	0.996	0.004	0.894	0.001	0.999	0.108	0.998	0.062	0.002	1652656
		3	0.959	0.929	0.943	0.998	0.002	0.929	0.001	0.999	0.071	0.999	0.041	0.001	915113
	3	1	0.833	0.749	0.784	0.987	0.013	0.755	0.006	0.994	0.251	0.993	0.167	0.007	9143203
		2	0.947	0.935	0.940	0.998	0.002	0.928	0.001	0.999	0.065	0.999	0.053	0.001	1000641
		3	0.983	0.981	0.982	0.999	0.001	0.978	0.000	1.000	0.019	1.000	0.017	0.000	477340

Table C.34: SBR all results for generated simulations based on following parameters: $\mathcal{P}=1$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$f$	$\$c$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time(ns)
25	1	1	0.196	1.000	0.328	0.819	0.181	0.397	0.189	0.811	0.000	1.000	0.804	0.000	152000
		2	0.497	1.000	0.658	0.935	0.065	0.660	0.068	0.932	0.000	1.000	0.503	0.000	77000
		3	0.575	1.000	0.726	0.950	0.050	0.719	0.052	0.948	0.000	1.000	0.425	0.000	66000
	2	1	0.673	1.000	0.802	0.968	0.032	0.790	0.034	0.966	0.000	1.000	0.327	0.000	42000
		2	0.907	1.000	0.950	0.992	0.008	0.942	0.009	0.991	0.000	1.000	0.093	0.000	21000
		3	0.974	1.000	0.986	0.998	0.002	0.984	0.002	0.998	0.000	1.000	0.026	0.000	26000
	3	1	0.869	1.000	0.928	0.989	0.011	0.919	0.011	0.989	0.000	1.000	0.131	0.000	32000
		2	0.981	1.000	0.990	0.998	0.002	0.988	0.002	0.998	0.000	1.000	0.020	0.000	19000
		3	0.993	1.000	0.996	0.999	0.001	0.995	0.001	0.999	0.000	1.000	0.007	0.000	23000
50	1	1	0.107	1.000	0.193	0.826	0.174	0.295	0.178	0.822	0.000	1.000	0.893	0.000	346000
		2	0.247	1.000	0.395	0.925	0.075	0.469	0.077	0.923	0.000	1.000	0.753	0.000	154000
		3	0.329	1.000	0.492	0.949	0.051	0.547	0.052	0.948	0.000	1.000	0.671	0.000	134000
	2	1	0.466	1.000	0.632	0.966	0.034	0.651	0.035	0.965	0.000	1.000	0.534	0.000	102000
		2	0.810	1.000	0.894	0.992	0.008	0.885	0.008	0.992	0.000	1.000	0.190	0.000	37000
		3	0.923	1.000	0.958	0.997	0.003	0.953	0.003	0.997	0.000	1.000	0.077	0.000	35000
	3	1	0.758	1.000	0.860	0.988	0.012	0.850	0.012	0.988	0.000	1.000	0.242	0.000	39000
		2	0.962	1.000	0.980	0.998	0.002	0.977	0.002	0.998	0.000	1.000	0.038	0.000	33000
		3	0.991	1.000	0.995	1.000	0.000	0.994	0.000	1.000	0.000	1.000	0.010	0.000	30000
75	1	1	0.076	1.000	0.141	0.826	0.174	0.249	0.176	0.824	0.000	1.000	0.924	0.000	592000
		2	0.164	1.000	0.281	0.925	0.075	0.385	0.076	0.924	0.000	1.000	0.836	0.000	221000
		3	0.267	1.000	0.419	0.947	0.053	0.483	0.053	0.947	0.000	1.000	0.733	0.000	165000
	2	1	0.382	1.000	0.549	0.969	0.031	0.588	0.032	0.968	0.000	1.000	0.618	0.000	113000
		2	0.760	1.000	0.860	0.992	0.008	0.851	0.008	0.992	0.000	1.000	0.240	0.000	44000
		3	0.877	1.000	0.933	0.996	0.004	0.925	0.004	0.996	0.000	1.000	0.123	0.000	40000
	3	1	0.667	1.000	0.797	0.988	0.012	0.791	0.012	0.988	0.000	1.000	0.333	0.000	71000
		2	0.943	1.000	0.970	0.998	0.002	0.966	0.002	0.998	0.000	1.000	0.057	0.000	43000
		3	0.986	1.000	0.993	1.000	0.000	0.992	0.000	1.000	0.000	1.000	0.014	0.000	35000
100	1	1	0.059	1.000	0.112	0.835	0.165	0.221	0.167	0.833	0.000	1.000	0.941	0.000	632000
		2	0.127	1.000	0.225	0.921	0.079	0.337	0.079	0.921	0.000	1.000	0.873	0.000	329000
		3	0.185	1.000	0.311	0.948	0.052	0.412	0.052	0.948	0.000	1.000	0.815	0.000	228000
	2	1	0.279	1.000	0.434	0.966	0.034	0.506	0.034	0.966	0.000	1.000	0.721	0.000	188000
		2	0.684	1.000	0.808	0.992	0.008	0.804	0.008	0.992	0.000	1.000	0.316	0.000	73000
		3	0.840	1.000	0.911	0.996	0.004	0.904	0.004	0.996	0.000	1.000	0.160	0.000	65000
	3	1	0.563	1.000	0.716	0.986	0.014	0.720	0.014	0.986	0.000	1.000	0.437	0.000	101000
		2	0.935	1.000	0.965	0.999	0.001	0.961	0.001	0.999	0.000	1.000	0.065	0.000	73000
		3	0.975	1.000	0.987	0.999	0.001	0.985	0.001	0.999	0.000	1.000	0.026	0.000	37000

Table C.35: SBR all results for generated simulations based on following parameters:
 $\mathcal{P}=2$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$f$	$\$c$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time(ns)
25	1	1	0.200	1.000	0.332	0.738	0.262	0.377	0.281	0.719	0.000	1.000	0.800	0.000	184243
		2	0.455	1.000	0.620	0.903	0.097	0.622	0.103	0.897	0.000	1.000	0.545	0.000	92044
		3	0.507	1.000	0.669	0.916	0.084	0.665	0.090	0.910	0.000	1.000	0.493	0.000	109096
	2	1	0.558	1.000	0.705	0.912	0.088	0.695	0.095	0.905	0.000	1.000	0.442	0.000	97512
		2	0.814	1.000	0.894	0.979	0.021	0.880	0.022	0.978	0.000	1.000	0.186	0.000	40690
		3	0.884	1.000	0.936	0.989	0.011	0.927	0.012	0.988	0.000	1.000	0.116	0.000	34822
	3	1	0.501	1.000	0.666	0.880	0.120	0.648	0.130	0.870	0.000	1.000	0.499	0.000	146921
		2	0.864	1.000	0.926	0.987	0.013	0.915	0.013	0.987	0.000	1.000	0.136	0.000	49789
		3	0.917	1.000	0.956	0.993	0.007	0.948	0.008	0.992	0.000	1.000	0.083	0.000	36529
50	1	1	0.111	1.000	0.200	0.676	0.324	0.271	0.337	0.663	0.000	1.000	0.889	0.000	489047
		2	0.212	1.000	0.349	0.864	0.136	0.420	0.140	0.860	0.000	1.000	0.788	0.000	239079
		3	0.282	1.000	0.438	0.908	0.092	0.496	0.095	0.905	0.000	1.000	0.718	0.000	169249
	2	1	0.245	1.000	0.393	0.843	0.157	0.443	0.164	0.836	0.000	1.000	0.755	0.000	289000
		2	0.544	1.000	0.698	0.967	0.033	0.703	0.034	0.966	0.000	1.000	0.456	0.000	91924
		3	0.750	1.000	0.854	0.987	0.013	0.845	0.014	0.986	0.000	1.000	0.250	0.000	46617
	3	1	0.414	1.000	0.564	0.895	0.105	0.582	0.109	0.891	0.000	1.000	0.586	0.000	220751
		2	0.741	1.000	0.849	0.985	0.015	0.836	0.015	0.985	0.000	1.000	0.259	0.000	69614
		3	0.886	1.000	0.939	0.995	0.005	0.930	0.005	0.995	0.000	1.000	0.114	0.000	62433
75	1	1	0.076	1.000	0.141	0.670	0.330	0.223	0.339	0.661	0.000	1.000	0.924	0.000	804640
		2	0.149	1.000	0.258	0.862	0.138	0.352	0.141	0.859	0.000	1.000	0.851	0.000	374813
		3	0.207	1.000	0.342	0.905	0.095	0.421	0.097	0.903	0.000	1.000	0.793	0.000	239480
	2	1	0.196	1.000	0.328	0.857	0.143	0.398	0.147	0.853	0.000	1.000	0.804	0.000	389833
		2	0.401	1.000	0.570	0.962	0.038	0.596	0.039	0.961	0.000	1.000	0.599	0.000	148415
		3	0.625	1.000	0.767	0.984	0.016	0.763	0.016	0.984	0.000	1.000	0.375	0.000	91018
	3	1	0.238	1.000	0.383	0.859	0.141	0.430	0.145	0.855	0.000	1.000	0.762	0.000	430211
		2	0.629	1.000	0.769	0.984	0.016	0.764	0.016	0.984	0.000	1.000	0.371	0.000	90904
		3	0.858	1.000	0.923	0.996	0.004	0.914	0.005	0.995	0.000	1.000	0.142	0.000	70158
100	1	1	0.060	1.000	0.113	0.682	0.318	0.201	0.325	0.675	0.000	1.000	0.940	0.000	1030129
		2	0.111	1.000	0.200	0.857	0.143	0.303	0.145	0.855	0.000	1.000	0.889	0.000	465595
		3	0.156	1.000	0.269	0.906	0.094	0.369	0.095	0.905	0.000	1.000	0.844	0.000	334137
	2	1	0.143	1.000	0.249	0.846	0.154	0.337	0.157	0.843	0.000	1.000	0.857	0.000	550904
		2	0.337	1.000	0.501	0.963	0.037	0.544	0.038	0.962	0.000	1.000	0.663	0.000	192749
		3	0.521	1.000	0.683	0.983	0.017	0.692	0.018	0.982	0.000	1.000	0.479	0.000	108482
	3	1	0.192	1.000	0.321	0.860	0.140	0.384	0.143	0.857	0.000	1.000	0.808	0.000	541813
		2	0.575	1.000	0.728	0.986	0.014	0.727	0.015	0.985	0.000	1.000	0.425	0.000	121246
		3	0.801	1.000	0.888	0.995	0.005	0.878	0.005	0.995	0.000	1.000	0.199	0.000	73137

Table C.36: SBR all results for generated simulations based on following parameters: $\mathcal{P}=3$, $\$d=11$, $\$b=1$, $\$B=2$, $\$F=10$, $\Phi=1$, $\Psi=0$, $\$t=1$ and $\$T=100$

$\$g$	$\$f$	$\$c$	PPV	TPR	F_{1sc}	ACC	ERR	MCC	FPR	TNR	FNR	NPV	FDR	FOR	Time(ns)
25	1	1	0.203	1.000	0.338	0.700	0.300	0.366	0.328	0.672	0.000	1.000	0.797	0.000	226287
		2	0.455	1.000	0.619	0.897	0.103	0.621	0.111	0.889	0.000	1.000	0.545	0.000	106635
		3	0.484	1.000	0.650	0.891	0.109	0.643	0.119	0.881	0.000	1.000	0.516	0.000	114783
	2	1	0.509	1.000	0.662	0.878	0.122	0.650	0.134	0.866	0.000	1.000	0.491	0.000	128724
		2	0.738	1.000	0.839	0.963	0.037	0.826	0.039	0.961	0.000	1.000	0.262	0.000	68210
		3	0.852	1.000	0.916	0.985	0.015	0.906	0.016	0.984	0.000	1.000	0.148	0.000	40736
	3	1	0.390	1.000	0.555	0.736	0.264	0.516	0.298	0.702	0.000	1.000	0.610	0.000	272426
		2	0.758	1.000	0.860	0.974	0.026	0.845	0.028	0.972	0.000	1.000	0.242	0.000	80674
		3	0.829	1.000	0.905	0.984	0.016	0.892	0.017	0.983	0.000	1.000	0.171	0.000	64638
50	1	1	0.115	1.000	0.207	0.546	0.454	0.244	0.481	0.519	0.000	1.000	0.885	0.000	891764
		2	0.204	1.000	0.338	0.810	0.190	0.398	0.199	0.801	0.000	1.000	0.796	0.000	310522
		3	0.258	1.000	0.408	0.870	0.130	0.464	0.136	0.864	0.000	1.000	0.742	0.000	194558
	2	1	0.188	1.000	0.315	0.685	0.315	0.348	0.335	0.665	0.000	1.000	0.812	0.000	500171
		2	0.473	1.000	0.631	0.949	0.051	0.644	0.053	0.947	0.000	1.000	0.527	0.000	117804
		3	0.642	1.000	0.775	0.976	0.024	0.770	0.024	0.976	0.000	1.000	0.358	0.000	87783
	3	1	0.318	1.000	0.454	0.761	0.239	0.466	0.253	0.747	0.000	1.000	0.682	0.000	453899
		2	0.513	1.000	0.676	0.961	0.039	0.676	0.040	0.960	0.000	1.000	0.487	0.000	135702
		3	0.788	1.000	0.879	0.989	0.011	0.868	0.011	0.989	0.000	1.000	0.212	0.000	90455
75	1	1	0.079	1.000	0.146	0.530	0.470	0.200	0.489	0.511	0.000	1.000	0.921	0.000	1610252
		2	0.138	1.000	0.243	0.804	0.196	0.328	0.202	0.798	0.000	1.000	0.862	0.000	475909
		3	0.184	1.000	0.310	0.872	0.128	0.390	0.131	0.869	0.000	1.000	0.816	0.000	297850
	2	1	0.142	1.000	0.248	0.688	0.312	0.303	0.324	0.676	0.000	1.000	0.858	0.000	779495
		2	0.272	1.000	0.425	0.925	0.075	0.478	0.077	0.923	0.000	1.000	0.728	0.000	231772
		3	0.446	1.000	0.614	0.968	0.032	0.633	0.032	0.968	0.000	1.000	0.554	0.000	135254
	3	1	0.142	1.000	0.247	0.604	0.396	0.277	0.413	0.587	0.000	1.000	0.858	0.000	1364899
		2	0.419	1.000	0.589	0.961	0.039	0.603	0.040	0.960	0.000	1.000	0.581	0.000	166658
		3	0.715	1.000	0.832	0.989	0.011	0.822	0.011	0.989	0.000	1.000	0.285	0.000	95038
100	1	1	0.061	1.000	0.116	0.540	0.460	0.179	0.474	0.526	0.000	1.000	0.939	0.000	2500439
		2	0.106	1.000	0.191	0.798	0.202	0.285	0.206	0.794	0.000	1.000	0.894	0.000	612724
		3	0.137	1.000	0.240	0.866	0.134	0.337	0.137	0.863	0.000	1.000	0.863	0.000	453216
	2	1	0.099	1.000	0.179	0.657	0.343	0.247	0.354	0.646	0.000	1.000	0.901	0.000	1400142
		2	0.226	1.000	0.367	0.926	0.074	0.433	0.075	0.925	0.000	1.000	0.774	0.000	279192
		3	0.372	1.000	0.540	0.967	0.033	0.573	0.034	0.966	0.000	1.000	0.628	0.000	140806
	3	1	0.104	1.000	0.188	0.594	0.406	0.237	0.418	0.582	0.000	1.000	0.896	0.000	1965743
		2	0.360	1.000	0.527	0.962	0.038	0.556	0.039	0.961	0.000	1.000	0.640	0.000	192750
		3	0.613	1.000	0.758	0.987	0.013	0.753	0.013	0.987	0.000	1.000	0.387	0.000	129861



Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br