Advances in Intelligent Systems and Computing 376

Amr Mohamed
Paulo Novais
António Pereira
Gabriel Villarrubia González
Antonio Fernández-Caballero *Editors*

Ambient Intelligence – Software and Applications

6th International Symposium on Ambient Intelligence (ISAmI 2015)

Springer

Advances in Intelligent Systems and Computing

Volume 376

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within "Advances in Intelligent Systems and Computing" are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

More information about this series at http://www.springer.com/series/11156

Amr Mohamed · Paulo Novais
António Pereira · Gabriel Villarrubia González
Antonio Fernández-Caballero
Editors

Ambient Intelligence - Software and Applications

6th International Symposium on Ambient Intelligence (ISAmI 2015)

Springer

Editors
Amr Mohamed
Computer Science and Engineering Department, College of Engineering
Qatar University
Doha
Qatar

Paulo Novais
Departamento de Informatica, ALGORITMI Centre
University of Minho
Braga
Portugal

António Pereira
Escola Superior de Tecnologia e Gestäo de Leiria
Instituto Politécnico de Leiria
Leiria
Portugal

Gabriel Villarrubia González
Departamento de Informática y Automática
University of Salamanca
Salamanca
Spain

Antonio Fernández-Caballero
Departamento de Sistemas Informáticos
University of Castilla-La Mancha
Albacete
Spain

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-19694-7 ISBN 978-3-319-19695-4 (eBook)
DOI 10.1007/978-3-319-19695-4

Library of Congress Control Number: 2015940728

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

This volume contains the proceedings of the 6th International Symposium on Ambient Intelligence (ISAmI 2015). The symposium was held in Salamanca, Spain during June 3–5 at the University of Salamanca.

ISAmI has been running annually and aiming to bring together researchers from various disciplines that constitute the scientific field of Ambient Intelligence to present and discuss the latest results, new ideas, projects and lessons learned, namely in terms of software and applications, and aims to bring together researchers from various disciplines that are interested in all aspects of this area.

Ambient Intelligence is a recent paradigm emerging from Artificial Intelligence, where computers are used as proactive tools assisting people with their day-to-day activities, making everyone's life more comfortable.

After a careful review, 27 papers from 10 different countries were selected to be presented in ISAmI 2015 at the conference and published in the proceedings. Each paper has been reviewed by, at least, three different reviewers, from an international committee composed of 74 members from 24 countries.

Acknowledgments

Special thanks to the editors of the workshops AIfES. Special Session on Ambient Intelligence for e-Healthcare.

We want to thank all the sponsors of ISAmI'15: IEEE Sección España, CNRS, AFIA, AEPIA, APPIA, AI*IA, and Junta de Castilla y León.

ISAmI would not have been possible without an active Program Committee. We would like to thank all the members for their time and useful comments and recommendations.

We would also like to thank all the contributing authors and the Local Organizing Committee for their hard and highly valuable work.

Your work was essential to the success of ISAmI 2015.

June 2015

Amr Mohamed
Paulo Novais
António Pereira
Gabriel Villarrubia González
Antonio Fernández-Caballero

Organization

Program Committee Chairs

Amr Mohamed (Chair)—College of Engineering, Qatar University (Qatar)
António Pereira (Chair)—Instituto Politécnico de Leiria (Portugal)

Program Committee

Adolfo Barroso Gallego—University of Salamanca
Alberto Lopez Barriuso—University of Salamanca
Alvaro Castro González—Universidad Carlos III de Madrid
Andreas Riener—Johannes Kepler University Linz, Institute for Pervasive Computing
Andrei Olaru—University Politehnica of Bucharest
Ângelo Costa—Universidade do Minho
Ansar-Ul-Haque Yasar—Universiteit Hasselt—IMOB
Antonio Fernández-Caballero—Universidad de Castilla-La Mancha
António Pereira—Escola Superior de Tecnologia e Gestão do IPLeiria
Balbo Flavien—ISCOD/Henri Fayol Institute
Bartolomeo Sapio—Fondazione Ugo Bordoni
Bogdan Kwolek—AGH University of Science and Technology
Carlos Ramos—Instituto Superior de Engenharia do Porto
Cecilio Angulo—Universitat Politcnica de Catalunya
Cesar Analide—University of Minho
Davide Carneiro—University of Minho
Eduardo Dias—CITI—FCT/UNL
Emmanuel Adam—University of Lille Nord de France
Enzo Pasquale Scilingo—University of Pisa
Fernando Silva—Department of Informatics Engineering; School of Technology and Management; Polytechnic Institute of Leiria, Portugal

Flavia Delicato—Federal University of Rio de Janeiro (UFRJ)
Florentino Fdez-Riverola—University of Vigo
Gabriel Villarrubia González—University of Salamanca
German Nemiorvskij—Albstadt-Sigmaringen University
Goreti Marreiros—ISEP/IPP-GECAD
Guillaume Lopez—Aoyama Gakuin University, College of Science and Technology
Habib M. Fardoun—King Abdulaziz University
Hans W. Guesgen—Massey University
Ilias Maglogiannis—University of Piraeus
Jaderick Pabico—University of the Philippines Los Banos
Javier Bajo—Universidad Politécnica de Madrid
Javier Carbo—Universidad Carlos III of Madrid
Javier Jaen—Polytechnic University of Valencia
Jean-Paul Jamont—LCIS, Université de Grenoble
Jesus Favela—CICESE
Jose Carlos Castillo Montoya—Universidad Carlos III de Madrid
Jose M. Molina—Universidad Carlos III de Madrid
José Machado—University of Minho, Computer Science and Technology Centre
Jose Manuel Pastor Garcia—Universidad de Castilla-La Mancha
José Ramón Álvarez-Sánchez—Universidad Nacional de Educación a Distancia
Joyca Lacroix—Philips Research
Juan Pavón—Universidad Complutense de Madrid
Julie Doyle—CASALA, Dundalk Institute of Technology
Junzhong Gu—East China Normal University
Koen van Dam—Imperial College London
Latif Ladid—UL & IPv6 Forum—IOT subC
Lawrence Wai-Choong Wong—National University of Singapore
Lourdes Borrajo—University of Vigo
Maria Navarro—University of Salamanca
Miguel A. Salichs—Universidad Carlos III de Madrid
Nadia Berthouze—University College London
Nuno M. Garcia—University of Beira Interior
Pablo Chamoso—University of Salamanca
Pascual González—Universidad de Castilla-La Mancha
Paulo Novais—University of Minho
Radu-Daniel Vatavu—University Stefan cel Mare of Suceava
Rafael Martinez Tomas—Universidad Nacional de Educación a Distancia
Remy Courdier—LIM—Université de la Réunion
Rene Meier—Lucerne University of Applied Sciences
Ricardo Costa—ESTGF.IPP
Rui José—University of Minho
S. Shyam Sundar—Penn State University, USA & Sungkyunkwan University, Korea
Sergio Sanchez González—University of Salamanca

Sotiris Nikoletseas—University of Patras and CTI
Soumya Kanti Datta—EURECOM
Stéphanie Combettes—IRIT—University of Toulouse
Sylvie Ratté—École de technologie supérieure
Teresa Romão—DI/FCT/UNL
Tibor Bosse—Vrije Universiteit Amsterdam
Valérie Camps—University of Toulouse—IRIT, France
Vicente Julian—GTI-IA DSIC UPV
Yi Fang—Purdue University

Organizing Committee Chairs

Paulo Novais—University of Minho (Portugal)
Gabriel Villarrubia González—University of Salamanca (Spain)

Local Organization Committee

Javier Bajo—Polytechnic University of Madrid (Spain)
Juan F. De Paz—University of Salamanca (Spain)
Fernando De la Prieta—University of Salamanca (Spain)
Sara Rodriguez—University of Salamanca (Spain)
Tiancheng Li—University of Salamanca (Spain)
Javier Prieto Tejedor—University of Salamanca (Spain)
Cesar Analide—University of Minho (Portugal)

Workshops

AIfeH.Special Session on Ambient Intelligence for e-Healthcare

Organizing Committee
Antonio Fernández-Caballero, University of Castilla-La Mancha, Spain
Pascual González, University of Castilla-La Mancha, Spain
Elena Navarro, University of Castilla-La Mancha, Spain

Contents

Using Evolutionary Algorithms to Personalize Controllers in Ambient Intelligence

Shu Gao and Mark Hoogendoorn

Abstract As users can have greatly different preferences, the personalization of ambient devices is of utmost importance. Several approaches have been proposed to establish such a personalization in the form of machine learning or more dedicated knowledge-driven learning approaches. Despite its huge successes in optimization, evolutionary algorithms (EAs) have not been studied a lot in this context, mostly because it is known to be a slow learner. Currently however, quite fast EA based optimizers exist. In this paper, we investigate the suitability of EAs for ambient intelligence.

Keywords Ambient intelligence · Evolutionary algorithms · Personalization · CMA-ES

1 Introduction

The rise of ambient intelligence is becoming more and more apparent in our daily lives: an increasing number of devices is surrounding us that perform all kinds of measurements and try to utilize this information in an intelligent way, for instance by controlling certain actuators or providing some form of feedback. In order for environments or devices to act sufficiently intelligent they need to be able to learn from the behavior of the user. Users can for instance have completely different preferences from each other, and hence, if only a single strategy would be deployed the system would never be effective and the user experience would be disappointing. In addition, devices need to learn how to cooperate with each other, and given the wealth of different devices on the market you cannot predefine the way in which they should.

S. Gao (✉) · M. Hoogendoorn
Department of Computer Science, VU University Amsterdam,
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
e-mail: s.gao@vu.nl

M. Hoogendoorn
e-mail: m.hoogendoorn@vu.nl

A. Mohamed et al. (eds.), *Ambient Intelligence - Software and Applications*,
Advances in Intelligent Systems and Computing 376,
DOI 10.1007/978-3-319-19695-4_1

Learning of preferences and learning how to establish effective cooperation between devices has been a subject of study in the field of ambient intelligence (or under its closely related fields such as pervasive computing and ubiquitous computing), see e.g. [1, 10, 15]. In [1] three stages of adaptation are identified: (1) the initial phase during which data is collected; (2) learning of behavior based upon the data collected, and (3) coping with dynamic environments. Mainly in the first stage hardly any machine learning approaches are appropriate as they hardly have any data/experiences to learn from, whereas this is a crucial phase. In that phase, the learning algorithm should learn on-the-fly. One of the problem solvers known to work well in nature, evolutionary algorithms (EAs), has not received a lot of attention in this domain, and in particular not for the subproblem which has just been described. Although EAs are mostly seen as slow optimizers, they have been shown to work very well for a range of optimization problems, see e.g. [6]. Furthermore, approaches such as genetic programming (cf. [2]) are highly suitable to generate sophisticated controllers.

In this paper, we explore the suitability of EAs for an ambient intelligence task thereby assuming no data being available up front. More precisely, we study a scenario where multiple (possibly heterogenous) devices need to be controlled in a simple way, thereby taking the preferences of multiple users into account. The rationale for starting with a simple scenario is that we want to explore whether EAs are able to solve a relatively simple problem in a suitable way before we move on to more complex problems. We use the state-of-the-art evolutionary optimizer, namely the CMA-ES [9]. Given the nature of devices in ambient intelligence, we use different variants of the algorithm: a centralized versus representing a single central controller and a number of distributed controllers repressing individual devices with their own controller. As evaluation criteria we measure the quality of the solutions found in terms of the percentage from the optimal solution as well as the time required to find a reasonable solution. We compare the outcome with simple benchmark algorithms such as hill climbing and simulated annealing.

This paper is organized as follows: first, we present related work in Sect. 2. Thereafter, in Sect. 3 we present the learning approach and the experimental setup is presented in Sect. 4. The results are presented and analyzed in Sect. 5. Finally, Sect. 6 concludes the paper.

2 Related Work

As said in the introduction, a lot of authors acknowledge the importance of machine learning techniques in ambient intelligence. An overview of useful techniques as well as examples of machine learning applications are given in [1]. In quite some approaches, a dataset for training is assumed. For example, Mozer et al. [12] use artificial neural networks in an AmI environment. Classification is implemented in an environment named 'SmartOffice' by Gal et al. [7]. On the other hand, reinforcement learning is another approach that does not need training data which is applied to ambient intelligence environment by Mozer [11]. There are examples in which EAs are applied in Ambient Intelligence. Doctor, Hagras and Caalghan [5] for example use Genetic Programming as a benchmark algorithm, stating that GPs are less suitable to use in an online fashion as they require many generations. A Genetic Algorithm is applied in [3] but again not in an online fashion. In [4] EAs are used to compose software around applications. Hence, one can see that there is some work which combines EAs with Ambient Intelligence, but none have judged whether such approach could be suitable to use in an online fashion where users provide feedback and act as a fitness function.

3 Approach

In our approach, we assume an environment in which multiple ambient devices are present that are equipped with sensors and actuators and have controllers that express their behavior, i.e. map sensory values to actions. The mapping between devices and controllers is left open: on the one extreme each device could have its own controller whereas on the other side of the spectrum there could be a single controller for all devices jointly. In the environment one or multiple users are present each having their own preference in particular situations. Here, a situation is a unique combination of sensory values or possibly a set of such combinations which all map to the same situation. Learning such a mapping could be another learning endeavor but in this initial exploration of EAs for ambient intelligence this is beyond our scope. The main goal of our research is to create controllers for the ambient devices that satisfy the user preferences best, a problem which we formulate as a maximization problem of the following function:

$$F = \sum_{\forall S:SIT} \sum_{\forall U:USER} user\_satisfaction(U, S, actions_{controllers}(S, U))$$

In other words, the controllers should, for all situations, find the set of actions that satisfy the users most. Since the user satisfaction can only be provided by the user itself, this entails that the user needs to be consulted every time a new controller is generated. A secondary goal is therefore also to minimize the number of evaluations required by the algorithm to find the solution to avoid bothering the user too much, and the user having to bare a lot of non-satisfactory solutions.

We assume that the controller is optimized for each situation separately. For each of such situations, a controller is represented by means of numerical values for each action it can perform. For binary actions, the possible values are clearly limited to 0 and 1 whereas for continuous actions (e.g. light intensity, sound volume) the action can take any value which is appropriate for the action. Table 1 shows an example of such a representation.

Table 1 Example representation for one controller for a single situation

a_1	a_2	a_3	a_4
1	0.5	0	0.25

In order to solve the problem we have now created, we use a variety of different approaches: state of the art EAs, including the CMA-ES and Cooperative Co-Evolution as well as benchmark algorithms including hill climbing and simulated annealing. An alternative would also be to use reinforcement learning, but given the scope of the scenario explored in this paper (see Sect. 4), this is not a suitable option and therefore not discussed in detail here. Each of these algorithms is explained in more detail below.

3.1 CMA-ES

The CMA-ES [9] is an evolutionary strategy. In general, EAs work with a population of individuals (in our case expressing the value for actions for a certain situation), of which individuals are selected for mutation (on a single individual) and crossover (combining two or more individuals), resulting in new individuals. Out of the total pool of individuals a new population is selected again, and this process continues until some termination criterion is reached. An evolutionary strategy is a variant in which the individuals are composed of a series of real numbers and the individuals also contain a dedicated field which determines the mutation probability, this field is also subject to evolution, and hence, the mutation probability self-adapts. In the CMA-ES a covariance matrix is used in order to improve the effectiveness of generating new individuals. Explaining all details of the algorithm is beyond the scope of this paper, the reader is referred to [9] for more details. We use the CMA-ES in two variants: (1) a single evolutionary loop in which a central controller simply determines the action for the actuators of all devices for this particular situation, and (2) a decentralized approach where each device has its own controller and CMA-ES population to evolve the controller. For the latter case, we use the cooperative co-evolutionary approach as proposed by Potter and De Jong [13]. Here, a single device is selected while fixing the controllers of the other devices to the best one found until then. Each of the individuals of the population of the selected device is then evaluation in conjunction with these best controllers, resulting in a fitness score. After that, the next device is selected, etc. Devices are selected in a round robin fashion. For non-continuous actions the real value is rounded to the nearest value (i.e. 0/1) during the evaluation phase.

3.2 Standard GA

Next to the CMA-ES we also try a simpler variant of an EA, namely the so-called “standard GA”, which, contrary to the sophisticated CMA-ES, consists of individuals that are composed of bits and uses less sophisticated operators. Combinations of bits can represent continuous actions for our case. Mutation takes place via simple bit-flips whereas crossover is done via selecting a crossover point and selecting the first part of one parent and the second part of the other. Selection is done by means of probabilities proportional to the fitness of the individual.

3.3 Hill Climbing and Simulated Annealing

For hill climbing we simply try a random step in either direction of the value for an action and perform an evaluation, the best solution (current, with the random step added, or deducted) is selected as the next controller. The process ends once the stopping condition has been met.

In the simulated annealing approach (see [14]), which works with the notion of the temperature of the process, a step is performed in the search space (i.e. for the action) which is equal to the temperature. The temperature function used is temperature function: $T = \frac{T_0}{log(k)}$ where T_0 is the initial temperature and k is number of steps. New solutions are accepted when they are an improvement, or, if not, they are selected with a probability $e^{\frac{\Delta C}{T}}$ which is dependent on the temperature of the process and the improvement made, ΔC. This scheme enables more exploration in the initial phase and more exploitation in the end of the process.

4 Experimental Setup

In this section, we describe the setup we have used to evaluate the algorithms that have been specified in Sect. 3. First, the case study is explained, followed by the precise setup of the experiments.

4.1 Case Study

As our case study, we focus on an office setting with lights that need to be controlled. As said, the focus is not so much on a complex scenario, but to study the potential of EAs for a relatively simple scenario. Figure 1 shows the scenario is more detail.

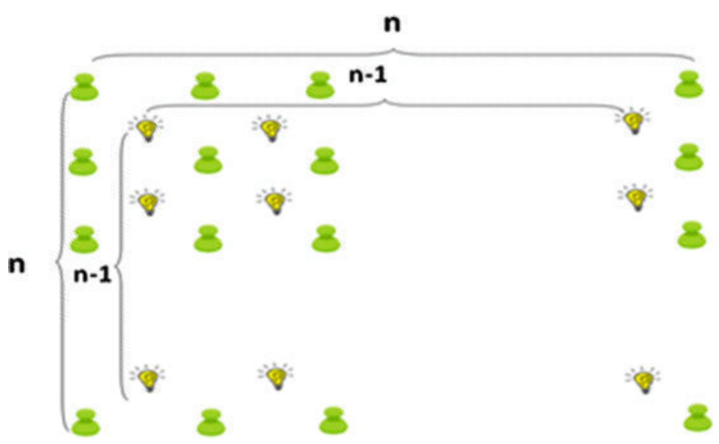

Fig. 1 Specific scenario

Essentially, there are more lights than users, and each user has a specific preference for a light intensity. The parameter n determines the complexity of the situation. Here, n defines the number of users (n^2) and the number of lights ($(n-1)^2$). We define such a complexity parameter as we want to study how the various approaches scale up with increasing complexity. We have chosen to have a number of lights which is smaller than the number of users to make the scenario more interesting. The intensity experienced by a user U is determined by all lights jointly, whereby the contribution of each individual light is determined by means of the following equation:

$$I(U) = \sum_{\forall L: LIGHTS} \frac{P(L)}{4\pi D(U,L)^2}$$

Here, *P(L)* is the power of the light L and *D(U, L)* is the distance of the light L to the user U. For now, we assume a single situation in which all users are present in the office. Each user has a preferred light intensity, thereby defining the function $user\_satisfaction(U, S, actions_{controllers}(S, U))$ as specified in Sect. 3:

$$user\_satisfaction(U, S, actions_{controllers}(S, U)) = |I(U) - pref\_int(U)|$$

4.2 Setup

We have implemented the entire system in Matlab, except of the CMA-ES which is available in C.[1] In our experiments, we run a number of different setups of the algorithm, in line with the approach outlined before, and which are shown in Table 2. CMA-ES is run with both a centralized and distributed controller setting, the standard GA only with a distributed setting, and the other benchmarks are only run in a centralized way. Note that the scenario does not contain any input states at the moment, making approaches such as reinforcement learning inappropriate. The precise algorithm parameters are expressed in the table as well.

We tried different levels of complexity ranging, namely n = 3,4,5,....,12, totaling to 10 scenarios. For each scenario we generate 10 instances with different preferences of users, and for each instance we perform 30 runs of the algorithms, given their probabilistic nature. In addition to the benchmarks indicated before, we also run an LP solver[2] to find the optimal solution to the problem. We assume that solutions within the range of 20 % from the optimal solution are satisfactory. As stopping criterion, the CMA-ES uses the fitness improvement as a metric, for the other algorithms the algorithm is stopped if it is within 20 % from the optimal solution or exceeds 100,000 fitness evaluations.

[1] https://www.lri.fr/~hansen/cmaesintro.html.

[2] http://lpsolve.sourceforge.net/5.5/.

Table 2 Experimental setup and settings

Abbreviation	Algorithm	Controller type	Specific settings
C-ES	CMA-ES	Centralized	Off-the-shelf toolkit, with population size set to $4 + (3 * log((n-1)^2))$
D-ES	CMA-ES	Distributed	See above, population size is set to 4 for each controller
D-GA	Standard GA	Distributed	Number of bits: 16 Population size: 100 Crossover rate: 0.6 Mutation rate: $\frac{1}{16}$
C-SA	Simulated annealing	Centralized	$T_0 = 100$
C-HC	Hill climbing	Centralized	Random number is selected from the range [0, 10]

5 Experimental Results

The experimental results are described in this section. First, we look at the number of evaluations needed to come to a reasonable solution (i.e. within 20 % from the optimal value). Figure 2 shows the results for the various algorithms. From the graph, it can be seen that a centralized controller generated by the CMA-ES algorithm by far outperforms the alternative algorithms, although the performance of the decentralized CMA-ES variant is still relatively close. The scaling of the algorithm seems good, given the exponential nature of the number of lights that needs to be controlled as a function of n on the x-axis. To be more precise, for the simple scenario, including 9 users and 4 lights, the system could find good solution within 200 evaluations. But it needs over 1500 evaluations in a complex scenario which involves 121 lights and 144 users. So, although from a scientific perspective the speedup is good, from a user perspective it is quite cumbersome. The variation of performance between the different runs is low for the CMA-ES. When we look at hill climbing and simulated annealing, we see that hill climbing performs a lot worse, with a huge variation. Simulated annealing does better, but does not come close to the speed of the CMA-ES variants. The Standard GA in the distributed setting is worst, most likely due to the distributed setting in combination with the simplicity of the EA. Table 3 shows the complete overview of the average times to find a solution with 20 % from optimal.

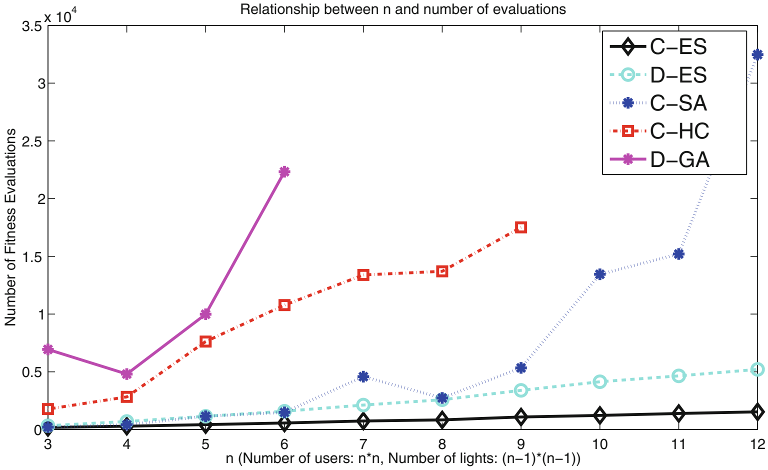

Fig. 2 Average time to optimal + 20 % for varying n

Furthermore, Fig. 3 shows the learning curve of the centralized CMA-ES for n = 3, it can be seen that the algorithm learns quite fast in the beginning, so the users are not exposed to very low quality solutions for a long period of time.

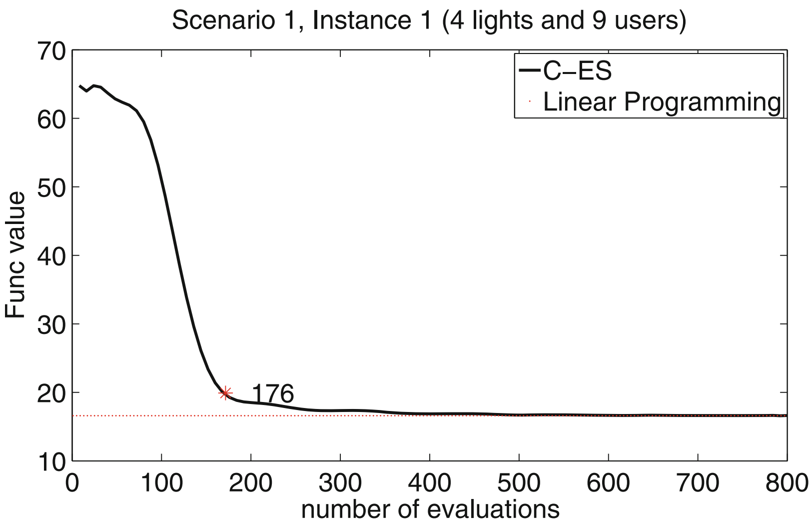

Fig. 3 The number of evaluations versus user satisfaction for C-ES

Table 3 Overview of the mean times to find a solution

Appr.	C-ES		D-ES		C-SA		C-HC		D-GA	
n	Ave	SD	Ave	SD	Ave	SD	Ave	SD	Ave	SD
3	179.2	28.9	333.6	54.3	206.8	107	1775.6	874.7	6941.8	10175.1
4	291	71.8	701.6	229.8	423	357.2	2828.3	2501.1	4806	3253.8
5	429.6	134.1	1148.8	443.7	1151	1789	7623.5	6169.5	9986.7	12361
6	569.4	87.9	1583.6	280.6	1484.3	919.2	10775.6	9118.9	22328.2	7644.7
7	743.4	206.8	2123.2	376.6	4580.3	7096.4	13388.7	12512.9	-	-
8	834	128.5	2584.4	352.7	2736.1	1572.1	13705.6	11093.5	-	-
9	1086.4	139.4	3396.8	350.3	5340.2	2325.6	17512.2	11957.3	-	-
10	1225.7	190.9	4140	282.8	13449.2	21352.6	-	-	-	-
11	1394	184.7	4648	401.6	15202.1	13940.4	-	-	-	-
12	1535.4	157.8	5206.8	552.9	32472.1	55847.1	-	-	-	-

6 Discussion

In this paper, we have explored the usability of EAs for personalization in ambient intelligence. Hereto, we have tried to formalize a fitness function, required for EAs, and have selected a first set of appropriate EA variants. In an experimental setting we have seen that EAs are able to find decent quality solutions, but as the problem becomes more complex the performance becomes a lot worse. Until now, the user feedback has just been the general level of satisfaction. Of course, more detailed feedback, or an initial phase of exploration could help to improve the speed to come to a solution and the quality of the solution. Ample approaches have utilized initial observations of users to derive a first set of reasonable controllers (see e.g. [8]). This was however not the purpose of this paper, we simply wanted to see whether an EA learning approach with one single piece of feedback could do the job, and the answer is that for simple environments this is possible, but as things get more complex this would become too much of a burden for users, let alone if multiple situations would need to be taken into account. Of course, the approach can still be applied, but our intuition is that one would need to resolve to alternative algorithms such as reinforcement learning.

For now an explicit fitness function in the form of user feedback has been obtained. We could also replace this with an alternative fitness function which is less direct (e.g. measure the work productivity), this would not change the setup of the learning system which shows how generic the approach is. How well the approach would learn the optimal lighting however would need to be studied, this would be an interesting aspect for future work. In addition, we want to explore more complex scenarios where sensors play a more prominent role and compare faster learning algorithms such as reinforcement learning to more knowledge driven approaches.

Acknowledgments This research is supported in part by scholarship from China Scholarship Council under number 201304910373. Furthermore, we would like to thank Gusz Eiben for the fruitful discussions and the anonymous reviewers for they valuable comments that helped to improve the paper.

References

1. A. Aztiria, A. Izaguirre, J.C. Augusto, Learning patterns in ambient intelligence environments: a survey. Artif. Intell. Rev. **34**(1), 35–51 (2010)
2. W. Banzhaf, P. Nordin, R. Keller, F. Francone, *Genetic Programming: An Introduction* (Morgan Kaufmann, San Francisco, 1998)
3. C.-K. Chiou, J.C. Tseng, G.-J. Hwang, S. Heller, An adaptive navigation support system for conducting context-aware ubiquitous learning in museums. Comput. Educ. **55**(2), 834–845 (2010)
4. O. Davidyuk, I. Selek, J. Imanol Duran, J. Riekki. Algorithms for composing pervasive applications. Int. J. Softw. Eng. Appl. **2**(2), 71–94 (2008)

5. F. Doctor, H. Hagras, V. Callaghan, A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **35**(1), 55–65 (2005)
6. A.E. Eiben, J. Smith, *Introduction to Evolutionary Computing* (Springer, London, 2003)
7. C.L. Gal, J. Martin, A. Lux, J.L. Crowley, Smart office: design of an intelligent environment. IEEE Intell. Syst. **16**(4), 60–66 (2001)
8. V. Guivarch, V. Camps, A. Peninou, Context Awareness in Ambient Systems by an Adaptive Multi-agent Approach, in *Ambient Intelligence*, vol. 7683, Lecture Notes in Computer Science, ed. by F. Paterno, B. de Ruyter, P. Markopoulos, C. Santoro, E. van Loenen, K. Luyten (Springer, Berlin, 2012), pp. 129–144
9. N. Hansen, A. Ostermeier, Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation, in *Proceedings of IEEE International Conference on Evolutionary Computation*, May 1996, pp. 312–317
10. F. Mattern, M. Naghshineh, in *Pervasive Computing (Proceedings of the Pervasive 2002 International Conference)*, number 2414 in LNCS. Springer, 2002. Online version of the proceedings at http://link.springer.de/link/service/series/0558/tocs/t2414.htm
11. M.C. Mozer, *Lessons from an Adaptive Home* (Wiley, New York, 2005), pp. 271–294
12. M.C. Mozer, R.H. Dodier, M. Anderson, L. Vidmar, R. F.C. III, D. Miller. The neural network house: an overview (1995)
13. M.A. Potter, K.A.D. Jong, A cooperative coevolutionary approach to function optimization, in *Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature*, PPSN III (Springer, London, UK, 1994), pp. 249–257
14. P. van Laarhoven, E. Aarts, Simulated annealing, in *Simulated Annealing: Theory and Applications, Mathematics and Its Applications*, vol. 37 (Springer, Netherlands, 1987), pp. 7–15
15. M. Weiser, Ubiquitous computing. Computer **26**(10), 71–72 (1993)

Automatic Early Risk Detection of Possible Medical Conditions for Usage Within an AMI-System

H. Joe Steinhauer and Jonas Mellin

Abstract Using hyperglycemia as an example, we present how Bayesian networks can be utilized for automatic early detection of a person's possible medical risks based on information provided by unobtrusive sensors in their living environments. The network's outcome can be used as a basis on which an automated AMI-system decides whether to interact with the person, their caregiver, or any other appropriate party. The networks' design is established through expert elicitation and validated using a half-automated validation process that allows the medical expert to specify validation rules. To interpret the networks' results we use an output dictionary which is automatically generated for each individual network and translates the output probability into the different risk classes (e.g., *no risk*, *risk*).

Keywords Ambient assisted living · Bayesian networks · Automated diagnosis

1 Introduction

A major part of the HELICOPTER (Healthy Life support through Comprehensive Tracking of individual and Environmental Behaviors, http://www.helicopter-aal.eu) is to develop information and communication technology (ICT) - based solutions that assist self-sufficient elderly people in early detection of the possible development of medical conditions, such as hyperglycemia or heart failure. The reason for this is to prevent complications arising from the medical conditions if they are not detected early enough. The main contribution of the HELICOPTER project is therefore the part of the system that can detect the risk of certain medical conditions

H. Joe Steinhauer (✉) · J. Mellin
University of Skövde, Skövde, Sweden
e-mail: joe.steinhauer@his.se

J. Mellin
e-mail: jonas.mellin@his.se

A. Mohamed et al. (eds.), *Ambient Intelligence - Software and Applications*,
Advances in Intelligent Systems and Computing 376,
DOI 10.1007/978-3-319-19695-4_2

based on sensor readings and that we call the *automatic triage*. Its system architecture is closer described in [1]. The automatic triage should be as unobtrusive as possible and should not bother the patient with unnecessary interventions. Health surveillance for the automatic triage is achieved by deploying unobtrusive sensors (e.g., infrared sensors, pressure sensors, power meters, body weight scales, and food-inventory tools) and wearable sensors (e.g., fall detectors, individual identification tags). All data collected from these heterogeneous sensors are then interpreted within a data analysis engine in order to deduce the patient's current risk of developing an acute medical condition (e.g., hyperglycemia or hypotension).

In this project it is our objective to utilize well established existing methods, in this case Bayesian networks, deploy them within a case study in order to develop the specific network designs necessary for each medical condition, and validate the resulting networks. The remainder of this paper is organized as follows: In Sect. 2 we explain how a Bayesian network for the use in the automatic triage can be developed in cooperation with a medical expert. After that, in Sect. 3, we describe how the results of Bayesian networks are validated. Last, but not least, we discuss our work and give some suggestions for future work in Sect. 4.

2 Bayesian Networks for Automatic Triage Diagnosis

Generally, a diagnosis will be determined on available evidence E and is defined as in e.g. [2]:

$$d^* = argmax_{d \in D} \Pr(d|E) \tag{1}$$

where D is the set of possible diagnoses, and d^* stands for the subset of diagnoses that have been chosen. Bayesian networks [3] have been used in the area of medical diagnostic reasoning, prognostic reasoning, treatment selection, and for the discovery of functional interactions, since the beginning of 1990 [2, 4, 5]. Some early examples can be found in [4, 6–8]. More recently, Bayesian networks are also applied in home care applications e.g. [9].

A Bayesian network [3] or causal probability network [6] is a graphical representation of a probability distribution over the set of random variables. Probabilistic inference can be done with Bayes rule (see e.g. [10]), which in our domain, where we want to infer the probability of a disease given that we observe one or several symptoms that are often caused by the disease, can be defined as:

$$P(disease|symptom) = \frac{P(symptom|disease)P(disease)}{P(symptom)} \tag{2}$$

Due to their graphical representation, Bayesian networks are relatively easy to understand and to create and can therefore be used, developed, and interpreted by domain experts [9]. They can often be seen as a model of cause-effect relationships [4] whereby their structure and the underlying probability distribution can be learnt from data or be created by hand. Thus qualitative and quantitative knowledge can be mixed [6]. Furthermore, uncertain knowledge can be modeled within a Bayesian network and missing data can be handled during the diagnosis process, which can successively be updated when more evidence becomes available [7].

Before we started to develop the automatic triage system, we also considered alternative evidential frameworks, such as evidence theory [11] and subjective logic [12], but decided together with the medical expert to use Bayesian networks based on four criteria: (1) the framework chosen needs to be able to express everything that is relevant for the task, (2) the design and inner workings of the framework should be easy to understand for the medical expert, (3) the framework should be considerably mature and (4) tools for developing the networks should be available.

In our project, as there is no data set available from that the Bayesian network could be automatically constructed and tested, it needs to be built by hand, whereby knowledge about the domain of diagnosing medical conditions is provided by a medical expert. [2] describes that the construction of a Bayesian network by hand usually involves five stages, which can be iterated during the construction process: (1) relevant variables need to be chosen; (2) relationships among the variables need to be identified; (3) logical and probabilistic constraints need to be identified and incorporated; (4) probability distributions need to be assessed; and (5) sensitivity analysis and evaluation of the network have to be performed.

Expert elicitation is an essential task in order to build the network and goes therefore hand in hand with the network construction. Following [13], expert elicitation is a five step process consisting of: (1) a decision has to be made how information will be used; (2) it has to be determined what information will be elicited from the expert; (3) the elicitation process needs to be designed; (4) the elicitation itself has to be performed; and (5) the elicited information needs to be translated (encoded) into quantities.

A specific problem when working with Bayesian networks is to elicit the prior and conditional probability values. [14] argue that even though probability theory is optimal for the task of decision making, it is often found to be impractical for people to use. On the other hand, qualitative approaches to deal with uncertainty, which appear to be more naturally usable by people, often lack in precision.

In order to elicit the prior and conditional probabilities for our project we developed a dictionary, which, as for example described in [14], can be specified to allow the expert to express his or her belief for or against a statement or claim in a so called argument. The argument is expressed in qualitative terms using qualifiers [14] that then are translated into probabilities. Several dictionaries have been described in the literature (e.g., [15]). However, for our task we needed to develop a suitable dictionary together with the expert, since it was important to the expert to know how the qualitative terms would translate into probabilities in order to fully understand what the qualitative terms stand for. It was also important that the