COPROD v2.0 13

Constraint Programming and Decision Making Workshop

Book of Abstracts

Martine Ceberio
Vladik Kreinovich

Organizers

The University of Texas at El Paso

November 1st, 2013

CoProD'13: 6th International Workshop on Constraint Programming and Decision Making
November 1st, 2013, at the University of Texas at El Paso
Chemistry and Computer Science Building
Room G.0208 (Ground Floor)

8:00 am	Registration and Breakfast
8:20 am – 8:30 am	Welcome
8:30 am – 9:25 am	First Invited Talk: Automatically verified computation with intervals, probability distributions and uncertain numbers *Scott Ferson, Applied Bioinformatics*
9:25 am – 9:50 am	An Approximation Method for a System of Mixed Equilibrium Problems *By B. Djafari-Rouhani, K. R. Kazmi, and S. H. Rizvi*
9:50 am – 10:15 am	Algebraic Product is the Only t-Norm for Which Optimization Under Fuzzy Constraints is Scale-Invariant *By Juan Carlos Figueroa, Universidad Nacional de Colombia en Bogota, Universidad Distrital de Bogota*
10:15 am – 10:30 am	Break
10:30 am – 10:55 am	Peak-End Rule: A Utility-Based Explanation *By Olga Kosheleva, Martine Ceberio, and Vladik Kreinovich, , University of Texas at El Paso*
10:55 am – 11:20 am	Similarity Approach to Defining Basic Level of Concepts Explained from the Utility Viewpoint *By Joe Lorkowski and Martin Trnecka, University of Texas at El Paso*
11:20 am – 1:00 pm	Lunch
1:00 pm – 1:55 pm	Second invited talk: **Linguistic Project Scheduling using Type-2 Fuzzy Sets** *By Juan Carlos Figueroa, Universidad Nacional de Colombia en Bogota, Universidad Distrital de Bogota*
1:55 pm – 2:20 pm	Towards a Physically Meaningful Definition of Computable Discontinuous and Multi-Valued Functions (Constraints) *By Martine Ceberio, Olga Kosheleva, and Vladik Kreinovich, University of Texas at El Paso*
2:20 pm – 2:45 pm	Energy-efficient Automatic Memory Management for Delay-Intolerant Systems *By Gabriel Arellano, Ed Dragone, Eric Freudenthal, Ed Hudgins, Md. Jahid, Roger Ochoa, David Pruitt, Adrian Veliz, University of Texas at El Paso*
2:45 pm – 3:00 pm	Break
3:00 pm – 3:25 pm	Predicting memory exhaustion by evaluating runtime features using machine learning *By Edward Hudgins, Gabriel Arellano, Eduardo Dragone, Eric Freudenthal, Olac Fuentes, M. D. Jahid, Rogelio Ochoa, David Pruitt, University of Texas at El Paso*
3:25 pm – 3:50 pm	Prioritizing Weak Reference Garbage Collection to Facilitate Object Caching *By Adrian Veliz and Eric Freudenthal, University of Texas at El Paso*
3:50 pm – 4:15 pm	Exploring the Effects of Emotive Stimuli in Presidential Policy Appeals *By José D. Villalobos and Cigdem V. Sirin, University of Texas at El Paso*
4:15 pm – 4:40 pm	How to Faster Test a Device for Different Combinations of Parameters *By Francisco Zapata and Luis Gutierrez, University of Texas at El Paso*
4:40 pm – 5:05 pm	Selecting Strategies Based on Abstracted Game Models *By Oscar Veliz and Christopher Kiekintveld, University of Texas at El Paso*
5:05 pm – 5:15 pm	Thanks and Concluding Remarks

Sponsored by: NSF

Hosted at: UTEP

Table of Contents

Automatically Verified Computation with Intervals, Probability Distributions and Uncertain Numbers

Invited Speaker: Scott Ferson

Applied Bioinformatics
100 North Country Road
Setauket, NY 11733 USA
scott@ramas.com

Abstract

Uncertainty propagation methods have often been disjoint and incomplete. Intervals alone cannot generally account for functional or stochastic dependence among variables, so propagations of interval uncertainty risk exploding to triviality. The dream of a workable "probabilistic arithmetic", imagined by many people, seems unachievable in practice. Whenever probability theory has been used to make calculations, analysts have routinely made untenable assumptions that ignore doubts about the model structure, the shape or precision of distribution specifications, and the character of stochastic dependence among variables. Until recently, such assumptions without any empirical justification have been common – even in relatively sophisticated and high-profile assessments such as risk analyses for space expeditions – because alternative methods that did not require these assumptions had not been available. New methods now allow us to compute often best-possible bounds on estimates of probabilities and probability distributions that are guaranteed to be correct even when one or more of the assumptions is relaxed or removed. This talk will present an overview of probability bounds analysis, as a computationally practical implementation of imprecise probabilities, that combines ideas from both interval analysis and probability theory to sidestep the limitations of each.

Invited Speaker's Bio

Scott Ferson is a senior scientist at Applied Biomathematics, a small-business research firm on Long Island, New York, and an adjunct at the School of Marine and Atmospheric Sciences. He holds a Ph.D. in ecology and evolution from Stony Brook University and has over 100 papers and 5 books on risk analysis and related topics. His recent work, funded primarily by NIH and NASA, has focused on developing statistical methods and software to solve quantitative assessment problems when data are poor or lacking and structural knowledge is severely limited.

Linguistic Project Scheduling using Type-2 Fuzzy Sets

Invited Speaker: Juan Carlos Figueroa

Universidad Distrital
Departamento de Ingenieria Industrial
Bogota, Colombia
jcfigueroag@udistrital.edu.co

Abstract

Project scheduling using PERT techniques are widely known among decision makers, due to its interpretability and simplicity. Sometimes, when starting a project, there is no complete information about the runtimes of the tasks of the project, so the analyst has to ask the experts of the project for information about them. The information provided by experts is in many cases based on its perception and expertise about each task, it is expressed using words instead of numbers, so there is a need for computing with words.

This way, Type-2 fuzzy sets seem to be an appropriate tool for dealing with uncertainty coming from perceptions and words, so what we propose is the use Type-2 fuzzy sets to represent the knowledge of multiple experts regarding the concept (word) of expected runtime of a task. Using linear programming models, we compute a set of possible Critical Paths that can happen depending on different uncertainty degrees, which are useful to see how the project does change in different pessimistic and optimistic scenarios.

Invited Speaker's Bio

Juan Carlos Figueroa-García was born in Bogotá - Colombia. He obtained a M.Sc.'s degree in Industrial Engineering at the Universidad Distrital Francisco José de Caldas in Bogotá in 2010, and currently is doing a Ph.D. studies in Engineering (optimization and operations research) at the National University of Colombia in Bogotá.

He has been an Assistant Professor in the Universidad Distrital Francisco José de Caldas in Bogotá since 2007, and a researcher in the National University of Colombia since 2010. His main interests are fuzzy linear programming, fuzzy statistics, fuzzy stochastic processes and mathematical modeling.

He is member of the IEEE Computational Intelligence Society, NAFIPS member, and ICIC (International Conference on Intelligent Computing) steering committee member. He has been awarded with the NAFIPS 2010 Outstanding paper award, the ICIC 2010 Best paper award, IFORS' grant for ELAVIO 2012, 2012 IFSA's Young scientist award, and NAFIPS 2013 Intervals session Best student paper award.

An Approximation Method for a System of Mixed Equilibrium Problems

B. Djafari-Rouhani[1], K. R. Kazmi[2], and S. H. Rizvi[2]

[1] Department of Mathematical Sciences
University of Texas at El Paso (UTEP),
500 W. University Ave., El Paso, Texas 79968, USA
behzad@utep.edu
[2] Department of Mathematics, Aligarh Muslim University
Aligarh 202002, India, krkazmi@gmail.com, shujarizvi07@gmail.com

Abstract. In this paper, we introduce an iterative method based on hybrid method, extragradient method and convex approximation method for finding a common element to the set of solutions of a system of unrelated equilibrium problems and the set of solutions to a common fixed-point problem for a family of nonexpansive mappings. We call it hybrid-extragradient-convex approximation method. We define the notion of a 2-monotone bifunction which is a natural extension of a 2-cyclically monotone operator. Further, we obtain a strong convergence theorem for the sequences generated by the proposed iterative scheme. Finally, we derive some consequences from our main result. The results presented in this paper extend and unify many of the previously known results in this area.

Keywords: System of unrelated equilibrium problems; common fixed-point problem; hybrid-extragradient-convex approximation method; 2-monotone bifunction; nonexpansive mapping; inverse-strongly monotone mapping; iterative scheme.

2000 Mathematics subject classifications: 49J30, 47H10, 47H17, 90C99

1 Introduction

Let H be real Hilbert space with inner product $\langle \cdot, \cdot \rangle$ and induced norm $\| \cdot \|$. For each $i = 1, 2, ..., N$, let K_i be a nonempty closed convex set with $\bigcap_{i=1}^{N} K_i \neq \emptyset$; let $F_i : K_i \times K_i \to \mathbb{R}$, where $\mathbb{R}$ is the set of real numbers, be a bifunction such that $F_i(x, x) = 0, \quad \forall x \in K_i$, and let $A_i : K_i \to H$ be a nonlinear mapping. Then, we consider the following new system of mixed equilibrium problems, which we call the system of unrelated mixed equilibrium problems (in short, SUMEP): Find $x \in \bigcap_{i=1}^{N} K_i$ such that

$$F_i(x, y_i) + \langle A_i x, y_i - x \rangle \geq 0, \quad \forall y_i \in K_i, \quad i = 1, 2, ..., N. \tag{1}$$

We note that for each $i = 1, 2, ..., N$, the mixed equilibrium problem (in short, MEP) [1] is to find $x_i \in K_i$ such that

$$F_i(x_i, y_i) + \langle A_i x_i, y_i - x_i \rangle \geq 0, \quad \forall y_i \in K_i, \quad i = 1, 2, ..., N. \tag{2}$$

We denote by $\text{MEP}(F_i, A_i, K_i)$, the set of solutions of MEP(2) corresponding to the mappings F_i, A_i and the set K_i. Then the set of solutions of SUMEP(1) is given by $\bigcap_{i=1}^{N} \text{MEP}(F_i, A_i, K_i)$. If $N = 1$, then SUMEP(1) is the well known mixed equilibrium problem (MEP) introduced by Moudafi and Théra [1]. If $F_i = 0$, SUMEP(1) reduces to the system of unrelated variational inequality problems (in short, SUVIP) considered and studied by Censor *et al.* [2] for set-valued version of mappings A_i. If $N = 1$ and $F_i = 0$, then SUMEP(1) reduces to the classical variational inequality problem (in short, VIP) introduced by Hartmann and Stampacchia [3]. If $A_i = 0$, SUMEP(1) reduces to the system of unrelated equilibrium problems (in short, SUEP) of finding $x \in \bigcap_{i=1}^{N} K_i$ such that

$$F_i(x, y_i) \geq 0, \quad \forall y_i \in K_i, \quad i = 1, 2, ..., N. \tag{3}$$

The set of solutions of SUEP(3) is denoted by $\text{EP}(F_i, K_i)$. If $N = 1$, SUEP(3) reduces to the well known equilibrium problem (in short, EP) introduced by Blum and Oettli [4]: Find $x \in K_1$ such that

$$F_1(x, y_1) \geq 0, \quad \forall y_1 \in K_1. \tag{4}$$

The solution set of EP(4) is denoted by $\text{EP}(F_1)$.

We also observe that if $F_i = 0$ and $A_i = 0$ for all i, then SUMEP(1) reduces to the problem of finding a point $x \in \bigcap_{i=1}^{N} K_i$ which is the well known convex feasibility problem (in short, CFP). If the sets K_i are fixed point sets of a family of operators $S_i : H \to H$, then the CFP is the common fixed point problem (in short, CFPP).

Recall that a mapping $S_i : K_i \to H$ is nonexpansive if

$$\|S_i x - S_i y\| \leq \|x - y\|, \quad \forall x, y \in K_i.$$

We denote the fixed point set of S_i by $\text{Fix}(S_i)$ for each $i = 1, 2, ...N$. We note that $\text{Fix}(S_i)$ is closed and convex, possibly empty (see e.g.[5]).

Motivated and inspired by the work of Nadezhkina and Takahashi [6], Peng and Yao [7], Censor *et al.* [2] and ongoing research in this direction, we introduce an iterative method based on hybrid method, extragradient method and convex approximation method for finding a common element of the set of solutions to SUMEP(1) and the set of solutions to CFPP of a family of nonexpansive mappings. We call it hybrid-extragradient-convex approximation method. We define the notion of a 2-monotone bifunction which is a natural extension of a 2-monotone operator. Further, we obtain a strong convergence theorem for the

sequences generated by the proposed iterative scheme. Then, we derive some consequences from our main result, which are also new. The result presented in this paper gives a unified treatment of some well known problems such as CFP, CFPP and SUVIP. The following is assumed in our main result.

Assumption 1 [4] *Let $F : C \times C \longrightarrow \mathbb{R}$ be a bifunction satisfying the following assumptions:*

(i) $F(x,x) = 0, \quad \forall x \in C$;
(ii) *F is monotone, i.e.,* $F(x,y) + F(y,x) \leq 0, \quad \forall x, y \in C$;
(iii) *For each $y \in C$, $x \to F(x,y)$ is hemi-uppersemicontimuous, i.e. for each $x, y, z \in C$,* $\limsup_{t \to 0^+} F(tz + (1-t)x, y) \leq F(x,y)$;
(iv) *For each $x \in C$, $y \to F(x,y)$ is convex and lower semicontinuous.*

2 Main Result

We prove the strong convergence of an iterative scheme based on hybrid method, extragradient method and convex approximation method which solves the problem of finding a common element to the solution set of SUMEP(1) and CFPP involving a family of nonexpansive mappings.

Theorem 1. *For each $i = 1, 2, ..., N$, let K_i be a nonempty closed convex subset of a real Hilbert space H with $\bigcap_{i=1}^{N} K_i \neq \emptyset$. Let $F_i : K_i \times K_i \to \mathbb{R}$ be a 2-monotone bifunction satisfying the Assumption 1, and the mapping $A_i : K_i \to H$ be σ_i-inverse strongly monotone. For each fixed i, let $S_i : K_i \to H$ be a nonexpansive mapping such that $\Omega = \bigcap_{i=1}^{N} (\mathrm{Fix}(S_i)) \bigcap \left(\bigcap_{i=1}^{N} \mathrm{MEP}(F_i, A_i, K_i) \right) \neq \emptyset$. Let the iterative sequences $\{x^n\}$, $\{y_i^n\}$ and $\{z_i^n\}$ be generated by the following iterative schemes:*

$$x^0 = x \in H,$$
$$y_i^n = T_{r_i^n}(x^n - r_i^n A_i x^n), \tag{5}$$
$$z_i^n = \alpha_i^n x^n + (1 - \alpha_i^n) S_i T_{r_i^n}(x^n - r_i^n A_i y_i^n), \tag{6}$$
$$C_i^n = \{z \in H : \|z_i^n - z\|^2 \leq \|x^n - z\|^2\}, \tag{7}$$
$$C^n = \bigcap_{i=1}^{N} C_i^n \tag{8}$$
$$Q^n = \{z \in H : \langle x^n - z, x - x^n \rangle \geq 0\}, \tag{9}$$
$$x^{n+1} = P_{C^n \cap Q^n} x, \tag{10}$$

for $n = 1, 2, ...,$ and for each $i = 1, 2, ..., N$ where $\{r_i^n\} \subset [a, b]$ for some $a, b \in (0, \sigma)$ and $\{\alpha_i^n\} \subset [0, c]$ for some $c \in [0, 1)$, where $\sigma := \min_{1 \leq i \leq N} \sigma_i$. Then the sequences $\{x^n\}$, $\{y_i^n\}$ and $\{z_i^n\}$ converge strongly to $d = P_\Omega x$.

The following corollary is due to [2] where A_i are single-valued mappings.

Corollary 1. [2] *For each $i = 1, 2, ..., N$, let K_i be a nonempty closed and convex subset of a real Hilbert space H with $\bigcap_{i=1}^{N} K_i \neq \emptyset$; let the mapping $A_i : K_i \to H$ be σ_i-inverse strongly monotone. Assume that $\Omega = \bigcap_{i=1}^{N} (\text{SUVIP}) \neq \emptyset$. Let the iterative sequences $\{x^n\}$, $\{y_i^n\}$ and $\{z_i^n\}$ be generated by the following iterative scheme:*

$$\begin{aligned}
x^0 &= x \in H, \\
y_i^n &= P_{K_i}(x^n - r_i^n A_i x^n) \\
z_i^n &= P_{K_i}(x^n - r_i^n A_i y_i^n), \\
C_i^n &= \{z \in H : \|z_i^n - z\|^2 \leq \|x^n - z\|^2\}, \\
C^n &= \bigcap_{i=1}^{N} C_i^n \\
Q^n &= \{z \in H : \langle x^n - z, x - x^n \rangle \geq 0\}, \\
x^{n+1} &= P_{C^n \cap Q^n} x,
\end{aligned}$$

for $n = 1, 2, ...,$ and for each $i = 1, 2, ..., N$, where $\{r_i^n\} \subset [a, b]$ for some $a, b \in (0, \sigma)$, where $\sigma = \min_{1 \leq i \leq N} \sigma_i$. Then the sequences $\{x^n\}$, $\{y_i^n\}$ and $\{z_i^n\}$ converge strongly to $d = P_\Omega x$.

3 Conclusion and Future Directions

We introduced an iterative scheme for finding a common element to the set of solutions of a system of unrelated equilibrium problems and the common fixed point set of a family of nonexpansive mappings, and proved the strong convergence of the sequences generated by the scheme to that element. Future directions to be pursued in the context of this research include the investigation of the problem when the operators A_i are set-valued operators as in [2], as well as the investigation of the convergence analysis for a regularized problem with perturbed data as in [9].

References

1. A. Moudafi and M. Théra, Proximal and dynamical approaches to equilibrium problems, Ill-posed variational problems and regularization techniques (Trier, 1998), 187–201, Lecture Notes in Econom. and Math. Systems, 477, Springer, Berlin, 1999.
2. Y. Censor, A. Gibali, S. Reich and S. Sabach, Common solutions to variational inequalities, Set-Valued Var. Anal. **20** (2012), 229–247.

3. P. Hartman and G. Stampacchia, On some non-linear elliptic differential-functional equations, Acta Math. **115** (1966), 271–310.
4. E. Blum and W. Oettli, From optimization and variational inequalities to equilibrium problems, Math. Student **63** (1994), 123–145.
5. K. Goebel and W. A. Kirk, Topics in metric fixed point theory, Cambridge Studies in Advanced Mathematics, 28, Cambridge University Press, Cambridge, 1990.
6. N. Nadezhkina and W. Takahashi, Strong convergence theorem by a hybrid method for nonexpansive mappings and Lipschitz continuous monotone mappings, SIAM J. Optim. **16** (2006), 1230–1241.
7. J. W. Peng and J. C. Yao, Some new extragradient-like methods for generalized equilibrium problems, fixed point problems and variational inequality problems, Optim. Methods Softw. **25** (2010), 677–698.
8. Z. Opial, Weak convergence of the sequence of successive approximations for nonexpansive mappings, Bull. Amer. Math. Soc. **73** (1967), 591–597.
9. F. Liu and M. Z. Nashed, Regularization of nonlinear ill-posed variational inequalities and convergence rates, Set-Valued Anal. **6** (1998), 313–344.
10. P. L. Combettes and S. A. Hirstoaga, Equilibrium programming in Hilbert spaces, J. Nonlinear Convex Anal. **6** (2005), 117–136.

Algebraic Product is the Only t-Norm for Which Optimization Under Fuzzy Constraints is Scale-Invariant

Juan Carlos Figueroa Garcia[1], Martine Ceberio[2], and Vladik Kreinovich[2]

[1] Universidad Distrital, Departamento de Ingenieria Industrial
Bogota, Colombia, filthed@gmail.com
[2] Department of Computer Science, University of Texas at El Paso
El Paso, TX 79968, USA, mceberio@utep.edu, vladik@utep.edu

Abstract. In many practical situations, we need to optimize under fuzzy constraints. There is a known Bellman-Zadeh approach for solving such problems, but the resulting solution, in general, depends on the choice of a not well-defined constant M. We show that this dependence disappears if we use an algebraic t-norm (and-operation) $f_{\&}(a,b) = a \cdot b$, and we also prove that the algebraic product is the only t-norm for which the corresponding solution is independent on M.

1 Formulation of the Problem

Need for optimization under fuzzy constraints. In decision making, we would like to find the best solution x among all possible solutions.

For example, if we need to build a chemical plant for producing chemicals needed for space exploration and for sophisticated electronics, then we need to select a design which is the most profitable among all the designs whose possible negative effect on the environment is small. In this example, the objective function is the overall profit.

In this example (and in many similar examples) the objective functions is well defined in the sense that for each alternative x, we can compute the exact value $f(x)$ of the objective function for this particular design. In contrast, the constraints are *not* well-defined, they are formulated by using words from a natural language (like "small"), words which are nor precise.

A reasonable way to describe the meaning of such imprecise ("fuzzy") constraints is to use techniques of fuzzy logic (see, e.g., [4, 6, 8]), where to each possible alternative x, we assign a number $\mu_c(x)$ describing to what extent this design satisfies the corresponding constraint. To find this value $\mu_c(x)$, we can, e.g., ask the user to mark this extent on a scale from 0 to 10, and if the user marks 7, take $\mu_c(x) = 7/10$.

This way, the original problem becomes a problem of optimization under fuzzy constraint: find x for which $f(x)$ is the largest possible among all x which satisfy the constraint described by a function $\mu_c(x)$.

Bellman-Zadeh approach to optimization under fuzzy constraints. To solve such problems, R. Bellman (a known specialist in optimization) and L. Zadeh (the founder of the fuzzy logic approach) came back with the following idea; see, e.g., [1, 4].

First, we (somehow) find the smallest value m of the objective function $f(x)$ among all possible solutions x, and we also find the largest possible value M of the objective function over all possible constraints. based on the values m and M, we can form, for each alternative x, the degree $\mu_m(x)$ to which x is maximal, as $\mu_m(x) \stackrel{\text{def}}{=} \dfrac{f(x)-m}{M-m}$. The larger $f(x)$, the larger this degree, and it reaches the value 1 if $f(x)$ attains the largest possible value M.

We want to find an alternative which satisfies the constraints *and* optimizes the objective function. In fuzzy techniques, the degree of truth in "and"-statement is approximately described by applying an appropriate *t-norm* $f(a,b)$ to the degrees to which both statements are true; see, e.g., [4, 6]. A t-norm must satisfy several natural properties: e.g., the fact that $A \,\&\, B$ means the same as $B \,\&\, A$ leads to the commutativity $f_{\&}(a,b) = f_{\&}(b,a)$, and the fact that "true" $\&\, A$ is equivalent to just A leads to the property $f_{\&}(1,a) = a$.

- By applying the t-norm $f_{\&}(a,b)$ to the degrees $\mu_c(x)$ and $\mu_m(x)$, we find the degrees $\mu_s(x) = f_{\&}(\mu_c(x), \mu_m(x))$ to which each alternative x is a solution.
- We then select the alternative which is the best fit, i.e., for which the degree $\mu_s(x)$ is the largest.

Problem: the value M is not well defined. Usually, we have some prior experience with similar problems, so we know some alternative(s) x which were previously selected. The value $f(x)$ for such "status quo" alternatives can be used as the desired minimum m.

Finding M is much more complicated, we do not know which alternatives to include and which not to include. If we replace the original value M with a new value $M' > M$, then the maximizing degree changes, from $\mu_m(x) = \dfrac{f(x)-m}{M-m}$ to $\mu'_m(x) = \dfrac{f(x)-m}{M'-m}$. One can easily see that $\mu'_m(x) = \lambda \cdot \mu_m(x)$ for $\lambda \stackrel{\text{def}}{=} \dfrac{M-m}{M'-m} < 1$.

The problem is that in general, the alternatives for which the functions $\mu_s(x) = f_{\&}(\mu_c(x), \mu_m(x))$ and $\mu'_s(x) = f_{\&}(\mu_c(x), \mu'_m(x)) = f_{\&}(\mu_c(x), \lambda \cdot \mu_m(x))$ may be different.

It is therefore desirable to come up with a scheme in which the solution would not change if we simply re-scale $\mu_m(x)$ by modifying the not well-defined quantity M.

What we do in this paper. In this paper, we show that the dependence on M disappears if we use algebraic product t-norm $f_{\&}(a.b) = a \cdot b$. We also show that this is the only t-norm for which decisions do not depend on M.

2 Main Results

Definition 1. *By a* t-norm*, we mean a function* $f_{\&} : [0,1] \times [0,1] \to [0,1]$ *for which* $f_{\&}(a,b) = f_{\&}(b,a)$ *and* $f_{\&}(1,a) = a$ *for all* a *and* b*.*

Comment. Usually, it is also required that the t-norm is associative. However, our results do not need associativity, so they are valid for non-associative and-operations as well; such non-associative operations are sometimes used to more adequately describe human reasoning; see, e.g., [2, 3, 5, 7, 9].

Definition 2. *Let* $f_{\&}(a,b)$ *be a t-norm. We say that optimization under fuzzy constraints is* scale-invariant *for this t-norm if for every set* X*, for every two functions* $\mu_c : X \to [0,1]$ *and* $\mu_m : X \to [0,1]$*, and for every real number* $\lambda \in (0,1)$*, we have* $S = S'$*, where:*

- S *is the set of all* $x \in X$ *for which the function* $\mu_s(x) = f_{\&}(\mu_c(x), \mu_m(x))$ *attains its maximum, i.e., for which* $\mu_s(x) = \max\limits_{y \in X} \mu_s(y)$*;*
- S' *is the set of all* $x \in X$ *for which the function* $\mu'_s(x) = f_{\&}(\mu_c(x), \lambda \cdot \mu_m(x))$ *attains its maximum, i.e., for which* $\mu'_s(x) = \max\limits_{y \in X} \mu'_s(y)$*.*

Proposition 1. *For the algebraic product t-norm* $f_{\&}(a,b) = a \cdot b$*, optimization under fuzzy constraints is scale-invariant.*

Proposition 2. *The algebraic product t-norm* $f_{\&}(a,b) = a \cdot b$ *is the only t-norm for which optimization under fuzzy constraints is scale-invariant.*

Proof of Proposition 1. For the algebraic product t-norm:

- S is the set of all $x \in X$ for which the function $\mu_s(x) = \mu_c(x) \cdot \mu_m(x)$ attains its maximum, and
- S' is the set of all $x \in X$ for which the function $\mu'_s(x) = \mu_c(x) \cdot \lambda \cdot \mu_m(x)$ attains its maximum.

Here, $\mu'_s(x) = \lambda \cdot \mu_s(x)$ for a positive number λ. Clearly, $\mu_s(x) \geq \mu_s(y)$ if and only if $\lambda \cdot \mu_s(x) \geq \lambda \cdot \mu_s(y)$, so the optimizing sets S and S' indeed coincide.

Proof of Proposition 2. Let $f_{\&}(a,b)$ be a t-norm for which optimization under fuzzy constraints is scale-invariant, and let a and b be two number from the interval $[0,1]$. Let us prove that $f_{\&}(a,b) = a \cdot b$.

Let us consider $X = \{x_1, x_2\}$ with $\mu_c(x_1) = \mu_m(x_2) = a$ and $\mu_c(x_2) = \mu_m(x_1) = 1$. In this case, $\mu_s(x_1) = f_{\&}(\mu_c(x_1), \mu_m(x_1)) = f_{\&}(a,1)$. Due to commutativity, we get $\mu_s(x_1) = f_{\&}(1,a)$ and due to the second property of the t-norm, we get $\mu_s(x_1) = a$.

Similarly, we have $\mu_s(x_2) = f_{\&}(\mu_c(x_2), \mu_m(x_2)) = f_{\&}(1,a)$. Due to the second property of the t-norm, we also get $\mu_s(x_2) = a$.

Since $\mu_s(x_1) = \mu_s(x_2)$, the optimizing set S consists of both elements x_1 and x_2.

Due to scale-invariance, for $\lambda = b$, the same set $S' = S = \{x_1, x_2\}$ must be the optimizing set for the function $\mu'_s(x) = f_{\&}(\mu_c(x), \lambda \cdot \mu_m(x))$. Thus, we must have $\mu'_s(x_1) = \mu'_s(x_2)$, i.e., $f_{\&}(a, b \cdot 1) = f_{\&}(1, b \cdot a)$. So, $f_{\&}(a, b) = f_{\&}(1, a \cdot b)$. Due to the second property of the t-norm, we conclude that $f_{\&}(a, b) = a \cdot b$.

The proposition is proven.

Acknowledgments. This work was supported in part by the National Science Foundation grants 0953339, HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, by Grants 1 T36 GM078000-01 and 1R43TR000173-01 from the National Institutes of Health, and by a grant N62909-12-1-7039 from the Office of Naval Research.

This work was performed when Juan Carlos Figueroa Garcia was a visiting researcher at the University of Texas at El Paso.

References

1. R. E. Bellman and L. A. Zadeh, "Decision making in a fuzzy environment", *Management Science*, 1970, Vol. 47, No. 4, pp. B141–B145.
2. B. Bouchon-Meunier, V. Kreinovich, and H. T. Nguyen, "Non-Associative Operations", *Proceedings of the Second International Conference on Intelligent Technologies InTech'2001*, Bangkok, Thailand, November 27–29, 2001, pp. 39–46.
3. I. R. Goodman, R. A. Trejo, V. Kreinovich, J. Martinez, and R. Gonzalez, "An even more realistic (non-associative) interval logic and its relation to psychology of human reasoning", *Proceedings of the Joint 9th World Congress of the International Fuzzy Systems Association and 20th International Conference of the North American Fuzzy Information Processing Society IFSA/NAFIPS'2001*, Vancouver, Canada, July 25–28, 2001, pp. 1586–1591.
4. G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, Upper Saddle River, New Jersey, 1995.
5. J. Martinez, L. Macias, A. Esper, J. Chaparro, V. Alvarado, S. A. Starks, and V. Kreinovich, "Towards more realistic (e.g., non-associative) and- and or-operations in fuzzy logic", *Proceedings of the 2001 IEEE Systems, Man, and Cybernetics Conference*, Tucson, Arizona, October 7–10, 2001, pp. 2187–2192.
6. H. T. Nguyen and E. A. Walker, *First Course In Fuzzy Logic*, CRC Press, Boca Raton, Florida, 2006.
7. R. Trejo, V. Kreinovich, I. R. Goodman, J. Martinez, and R. Gonzalez, "A Realistic (Non-Associative) Logic And a Possible Explanations of 7±2 Law", *International Journal of Approximate Reasoning*, 2002, Vol. 29, pp. 235–266.
8. L. A. Zadeh, "Fuzzy sets", *Information and Control*, 1965, Vol. 8, pp. 338–353.
9. H. H. Zimmerman and P. Zysno, "Latent connectives in human decision making", *Fuzzy Sets and Systems*, 1980, Vol. 4, pp. 37–51.

Peak-End Rule: A Utility-Based Explanation

Olga Kosheleva, Martine Ceberio, and Vladik Kreinovich

University of Texas at El Paso, El Paso, TX 79968, USA
{olgak,mceberio,vladik}@utep.edu

Abstract. In many practical situations, people judge their overall experience by only taking into account the peak and the last levels of pleasantness or unpleasantness. While this peak-end rule is empirically supported by numerous psychological experiments, it seems to contradict our general theoretical ideas about people's preferences. In this paper, we show that, contrary to this impression, the end-peak rule can be justified based on the main ideas of the traditional utility-based decision theory.

1 Peak-End Rule: Description and Need for an Explanation

Peak-end rule: empirical fact. In many situations, people judge their overall experience by the peak and end pleasantness or unpleasantness, i.e., by using only the maximum (minimum) and the last value; see, e.g., [1, 4].

This is true for people's perception of the unpleasantness of a medical procedure, of the quality of the cell phone perception, etc.

Need for an explanation. There is a lot of empirical evidence supporting the peak-end rule, but not much of an understanding. However, at first glance, the rule appears somewhat counter-intuitive: why only peak and last value? why not some average? In this paper, we provide such an explanation based on the traditional decision making theory.

2 Towards an Explanation

Traditional decision making theory: a brief reminder of utility approach. Our objective is to describe the peak-end rule in terms of the traditional decision making theory. According to decision theory, preferences of rational agents can be described in terms of *utility* (see, e.g., [2, 3]): a rational agent selects an action with the largest value of expected utility.

Utility is not uniquely defined. Utility is usually defined modulo a linear transformation. In the above experiments, we usually have a fixed *status quo* level which can be taken as 0. Once we fix this value at 0, the only remaining non-uniqueness in describing utility is scaling $u \to k \cdot u$.

Need for a utility-averaging operation. We want to describe the "average" utility corresponding to a sequence of different experiences. We assume that we know

the utility corresponding to each moment of time. To get an average utility value, we need to combine these momentous utilities into a single average.

If we have already found the average utility corresponding to two consequent sub-intervals of time, we then need to combine these two averages into a single average corresponding to the whole interval. In other words, we need an operation $a*b$ that, given the average utilities a and b corresponding to two consequent time intervals, generates the average utility of the combined two-stage experience.

Natural properties of the utility-averaging operation.

1) If we had the same average utility level $a = b$ on both stages, then this same value should be the two-stage average, i.e., we should have $a * a = a$. In mathematical terms, this means that the utility-averaging operation $*$ should be *idempotent*.

2) If we make one of the stages better, then the resulting average utility should increase (or at least not decrease) as well. In other words, the utility-averaging operation $*$ should be *monotonic* in the sense that if $a \le a'$ and $b \le b'$ then $a * b \le a' * b'$.

3) Small changes in one of the stages should lead to small changes in the overall average utility; in precise terms, this means that the function $a * b$ must be *continuous*.

4) For a three-stage situation, with average utilities a, b, and c corresponding to the three stages, we can compute the average utility in two different ways:

- we can first combine the utilities of the first two stages into an average value $a * b$, and then combine this average with c, resulting in $(a * b) * c$;
- alternatively, we can first combine the utilities b and c into $b * c$, and then combine a with $b * c$, resulting in $a * (b * c)$.

The resulting three-stage average should not depend on the order in which we combined the stages, so we should have $(a * b) * c = a * (b * c)$; in mathematical terms, the operation $a * b$ must be *associative*.

5) Finally, since utility is defined modulo scaling, it is reasonable to require that the utility-averaging operation does not change with scaling:

- In the original scale, we combine a and b and get $a * b$. In the new scale corresponding to a factor $k > 0$, this combined value has the form $k \cdot (a * b)$.
- After re-scaling, the original utilities get the new values $a' = k \cdot a$ and $b' = k \cdot b$. Averaging these two values leads to $a' * b' = (k \cdot a) * (k \cdot b)$ in the new scale.

The resulting average should not depend on how we deduced it, i.e., we should have $(k \cdot a) * (k \cdot b) = k \cdot (a * b)$ for all k, a and b.

What we plan to do. Let us show that the above reasonable requirements largely explain the peak-end phenomenon.

3 Main Result

Proposition 1. *Let $a * b$ be a binary operation on the set of all non-negative numbers which satisfies the following properties:*

1) *it is idempotent, i.e., $a * a = a$ for all a;*
2) *it is monotonic, i.e., $a \le a'$ and $b \le b'$ imply that $a * b \le a' * b'$;*
3) *it is continuous as a function of a and b;*
4) *it is associative, i.e., $(a * b) * c = a * (b * c)$;*
5) *it is scale-invariant, i.e., $(k \cdot a) * (k \cdot b) = k \cdot (a * b)$ for all k, a and b.*

Then, this operation coincides with one of the following four operations:

- $a_1 * \ldots * a_n = \min(a_1, \ldots, a_n)$;
- $a_1 * \ldots * a_n = \max(a_1, \ldots, a_n)$;
- $a_1 * \ldots * a_n = a_1$;
- $a_1 * \ldots * a_n = a_n$.

Comment. Thus, every utility-averaging operation which satisfies the above reasonable properties means that we select either the worst or the best or the first or the last utility. This (almost) justifies the peak-end phenomenon, with the only exception that in addition to peak and end, we also have the start $a_1 * \ldots * a_n = a_1$ as one of the options.

Proof.

1°. For every $a \ge 1$, let us denote $a * 1$ by $\varphi(a)$. For $a = 1$, due to the idempotence, $\varphi(1) = 1 * 1 = 1$. Due to monotonicity, $a \le a'$ implies that $\varphi(a) \le \varphi(a')$, i.e., that the function $\varphi(a)$ is (non-strictly) increasing.

2°. Due to associativity, for every a, we have $(a * 1) * 1 = a * (1 * 1)$. Due to idempotence, $1 * 1 = 1$, so the above equality takes the form $(a * 1) * 1 = a * 1$, i.e., the form $\varphi(\varphi(a)) = \varphi(a)$. Thus, for every value t from the range of the function $\varphi(a)$ for $a \ge 1$, we have $\varphi(t) = t$.

3°. Since the operation $a * b$ is continuous, the function $\varphi(a) = a * 1$ is also continuous. Thus, its range $S \stackrel{\text{def}}{=} \varphi([1, \infty))$ for $a \in [1, \infty)$ is a connected set, i.e., an interval (finite or infinite). Since the function $\varphi(a)$ is monotonic, and $\varphi(1) = 1$, this interval must start with 1. So, we have three possible options:

- $S = \{1\}$;
- $S = [1, k]$ or $S = [1, k)$ for some $k \in (1, \infty)$;
- $S = [1, \infty)$.

Let us consider these three options one by one.

3.1°. In the first case, $\varphi(a) = a * 1 = 1$ for all a. From scale invariance, we can now conclude that for all $a \ge b$, we have $a * b = b \cdot \left(\frac{a}{b} * 1\right) = b \cdot 1 = b$.

3.2°. In the second case, every value t between 1 and k is a possible value of $\varphi(a)$, thus $\varphi(t) = t * 1 = t$ for all such values t. In particular, for every $\varepsilon > 0$, for the value $t = k - \varepsilon$, we have $\varphi(k - \varepsilon) = k - \varepsilon$. Due to monotonicity, the value

$\varphi(k)$ must be not smaller than all these values $k-\varepsilon$, hence not smaller than k. On the other hand, all the values $\varphi(a)$ are less than or equal than k, so we must have $\varphi(k)=k$ as well. Similarly, for values $t \geq k$, due to monotonicity, we have $\varphi(t) \geq k$ and since always $\varphi(t) \leq k$, we conclude that $\varphi(t)=k$ for all $t \geq k$. Now, due to associativity, we have

$$((k-\varepsilon)^2 * (k-\varepsilon)) * 1 = (k-\varepsilon)^2 * ((k-\varepsilon) * 1). \tag{1}$$

Here, due to scale-invariance,

$$(k-\varepsilon)^2 * (k-\varepsilon) = (k-\varepsilon) \cdot ((k-\varepsilon) * 1) = (k-\varepsilon) \cdot \varphi(k-\varepsilon) =$$

$$(k-\varepsilon) \cdot (k-\varepsilon) = (k-\varepsilon)^2, \tag{2}$$

and therefore,

$$((k-\varepsilon)^2 * (k-\varepsilon)) * 1 = (k-\varepsilon)^2 * 1 = \varphi((k-\varepsilon)^2).$$

For $k > 1$, we have $k^2 > k$ and thus, for sufficiently small $\varepsilon > 0$, we have $(k-\varepsilon)^2 > k$. So, $\varphi((k-\varepsilon)^2) = k$, i.e., the left-hand side of the equality (1) is equal to k.

Let us now compute the right-hand side of the equality (1). Here, $(k-\varepsilon)*1 = k-\varepsilon$ and thus, the right-hand side has the form $(k-\varepsilon)^2 * (k-\varepsilon)$ which, as we already know (Equation (2)), is equal to $(k-\varepsilon)^2$. We already know that the left-hand side is equal to k, and that $(k-\varepsilon)^2 > k$. Thus, the equality (1) cannot be satisfied. This proves that the second case is impossible.

3.3°. In the third case, every value $t \geq 1$ is a possible value of $\varphi(a)$, thus

$$\varphi(t) = t * 1 = t$$

for all values $t \geq 1$. Thus, for all $a \geq b$, we have $a * b = b \cdot \left(\frac{a}{b} * 1\right) = b \cdot \frac{a}{b} = a$.

4°. Due to Part 3 of this proof, we have one of the following two cases:

- $\geq_1$: for all $a \geq b$, we have $a * b = b$;
- $\geq_2$: for all $a \geq b$, we have $a * b = a$.

Similarly, by considering $a \leq b$, we conclude that in this case, we also have two possible cases:

- $\leq_1$: for all $a \leq b$, we have $a * b = b$;
- $\leq_2$: for all $a \leq b$, we have $a * b = a$.

By combining each of the $\geq$ cases with each of the $\leq$ cases, we get the following four combinations:

- $\geq_1, \leq_1$: in this case, $a * b = b$ for all a and b, and therefore, $a_1 * \ldots * a_n = a_n$;
- $\geq_1, \leq_2$: in this case, $a * b = \min(a, b)$ for all a and b, and therefore,

$$a_1 * \ldots * a_n = \min(a_1, \ldots, a_n);$$

$\geq_2, \leq_1$: in this case, $a * b = \max(a, b)$ for all a and b, and therefore,

$$a_1 * \ldots * a_n = \max(a_1, \ldots, a_n);$$

$\geq_2, \leq_2$: in this case, $a * b = a$ for all a and b, and therefore, $a_1 * \ldots * a_n = a_1$.

The proposition is proven.

Case of negative utilities. The above formula shows how to combine positive experiences. A similar result can be proven for situations in which we need to combine unpleasant experiences, i.e., experience corresponding to negative utilities; the proof of this result is similar to the proof of Proposition 1.

Remaining open problems. Following the psychological experiments, we only considered the case when all experiences are positive and the case when all experiences are negative. What happens in the general case? If we impose an additional requirement of shift-invariance $(a + u_0) * (b + u_0) = a * b + u_0$, then we can get a result similar to Proposition 1 for this general case as well. But what if we do not impose this additional requirement?

Are all five conditions in Proposition 1 necessary? Some are necessary:

1) $a * b = a + b$ satisfies all the conditions except for idempotence;
4) $a * b = \dfrac{a + b}{2}$ satisfies all the conditions except for associativity;
5) the operation $a * b$ that returns the value from the interval $[\min(a, b), \max(a, b)]$ which is the closest to 1 satisfies all the conditions except for scale invariance.

However, it is not clear whether monotonicity and continuity are needed to prove our results.

Comment. In analyzing the need for these conditions, it may help to know that the set $\{z : z * 1 = z\}$ is a semigroup: indeed, if $z_1 * 1 = z_1$ and $z_2 * 1 = z_2$, then $(z_1 \cdot z_2) * (z_1 * 1) = (z_1 \cdot z_2) * z_1 = z_1 \cdot (z_2 * 1) = z_1 \cdot z_2$ and $((z_1 \cdot z_2) * z_1) * 1 = (z_1 \cdot z_2) * 1$, so associativity implies that $(z_1 \cdot z_2) * 1 = z_1 \cdot z_2$.

Acknowledgments. This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, by Grants 1 T36 GM078000-01 and 1R43TR000173-01 from the National Institutes of Health, and by a grant N62909-12-1-7039 from the Office of Naval Research.

References

1. Kahneman, D.: Objective happiness, in: Kahneman, D., Diener, E., and Schwarz, N. (eds.), Well-Being: The Foundations of Hedonistic Psychology, Russell Sage, New York (1999), pp. 3–25.
2. Luce, R. D., Raiffa, R.: Games and Decisions: Introduction and Critical Survey, Dover, New York (1989)
3. Raiffa, H.: Decision Analysis, Addison-Wesley, Reading, Massachusetts (1970)
4. McFadden, D. L.: The New Science of Pleasure, National Bureau of Economic Research, Working Paper 18687 (2013)
5. Bishop, E., Bridges, D. S.: Constructive Analysis, Springer, N.Y. (1985)

Similarity Approach to Defining Basic Level of Concepts Explained from the Utility Viewpoint

Joe Lorkowski[1] and Martin Trnecka[2]

[1] Department of Computer Science, University of Texas at El Paso
El Paso, TX 79968, USA, lorkowski@computer.org
[2] Department of Computer Science, Palacky University
Olomouc, Czech Republic, martin.trnecka@gmail.com

Abstract. In many practical situations, it is necessary to describe an image in words. From the purely logical viewpoint, to describe the same object, we can use concepts of different levels of abstraction: e.g., when the image includes a dog, we can say that it is a dog, or that it is a mammal, or that it is a German Shepherd. In such situations, humans usually select a concept which, to them, in the most natural; this concept is called the *basic level* concept. However, the notion of a basic level concept is difficult to describe in precise terms; as a result, computer systems for image analysis are not very good in selecting concepts of basic level. At first glance, since the question is how to describe human decisions, we should use notions from a (well-developed) decision theory – such as the notion of utility. However, in practice, a well-founded utility-based approach to selecting basic level concepts is not as efficient as a purely heuristic "similarity" approach. In this paper, we explain this seeming contradiction by showing that the similarity approach can be actually explained in utility terms – if we use a more accurate description of the utility of different alternatives.

1 Formulation of the Problem

What are basic level concepts and why their are important. With the development of new algorithms and faster hardware, computer systems are getting better and better in analyzing images. Computer-based systems are not yet perfect, but in many cases, they can locate human beings in photos, select photos in which a certain person of interest appears, and perform many other practically important tasks.

In general, computer systems are getting better and better in performing well-defined image understanding tasks. However, such systems are much less efficient in more open-ended tasks, e.g., when they need to describe what exactly is described by a photo.

For example, when we present, to a person, a photo of a dog and ask: "What is it?", most people will say "It is a dog". This answer comes natural to us, but, somewhat surprisingly, it is very difficult to teach this answer to a computer. The problem is that from the purely logical viewpoint, the same photo can be

characterized on a more abstract level ("an animal", "a mammal") or on a more concrete level ("German shepherd"). In most situations, out of many possible concepts characterizing a given object, concepts of different levels of generality, humans select a concept of a certain intermediate level. Such preferred concepts are known as *basic level* concepts.

We need to describe basic level concepts in precise terms. Detecting basic level concepts is very difficult for computers. The main reason for this difficulty is that computers are algorithmic machines. So, to teach computers to recognize basic level concepts, we need to provide explain this notion in precise terms – and we are still gaining this understanding.

Current attempts to describe basic level concepts in precise terms: a brief description. When we see a picture, we make a decision which of the concepts to select to describe this picture. In decision making theory, it is known that a consistent decision making can be described by *utility theory*, in which to each alternative A, we put into correspondence a number $u(A)$ called its *utility* in such a way that a utility of a situation in which we have alternatives A_i with probabilities p_i is equal to $\sum p_i \cdot u(A_i)$; see, e.g., [4, 5, 8, 10, 12].

Naturally, researchers tried to use utility theory to explain the notion of basic level concepts; see, e.g., [3, 7, 14]. In this approach, researchers analyze the effect of different selections on the person's behavior, and come up with the utility values that describes the resulting effects. The utility-based approach describes the basic level concepts reasonably well, but not perfectly. Somewhat surprisingly, a different approach – called *similarity approach* – seem to be more adequate in describing basic level concepts. The idea behind this approach was proposed in informal terms in [13] and has been described more formally in [11]. Its main idea is that in a hierarchy of concepts characterizing a given object, a basic level concept is the one for which the degree of similarity between elements is much higher than for the more abstract (more general) concepts and slightly smaller than for the more concrete (more specific) concepts. For example, we select a dog as a basic level concept because the degree of similarity between different dogs is much larger than similarity between different mammals – but, on the other hand, the degree of similarity between different German Shepherds is not that much higher than the degree of similarity between dogs of various breeds.

In our papers [1, 2], we transformed somewhat informal psychological ideas into a precise algorithms and showed that the resulting algorithms are indeed good in detecting basic level concepts.

Challenging question. From the pragmatic viewpoint, that we have an approach that works well is good news. However, from the methodological viewpoint, the fact that a heuristic approach works better than a well-founded approach based on decision theory – which describes rational human behavior – is a challenge.

What we do in this paper: main result. In this paper, we show – on the qualitative level – that the problem disappears if we describe utility more ac-

curately: under this more detailed description of utility, the decision-making approach leads to the above-mentioned similarity approach.

What we do in this paper: auxiliary result. It is usually more or less clear how to define degree of similarity – or, equivalent, degree of dissimilarity ("distance" $d(x, y)$) between two objects. There are several possible approaches to translate this distance between *objects* into distance between *concepts* (classes of objects). We can use worst-case distance $d(A, B)$ defined as the maximum of all the values $d(x, y)$ for all $x \in A$ and $y \in B$. Alternatively, we can use average distance as the arithmetic average of all the corresponding values $d(x, y)$. In [1], we compared these alternatives; it turns out that the average distance leads to the most adequate description of the basic level concepts.

In this paper, we provide a (qualitative) explanation of this empirical fact as well.

2 Analysis of the Problem and the Resulting Solution

What is the utility associated with concepts of different levels of generality. In the ideal world, when we make a decision in a certain situation, we should take into account all the information about this situation, and we should select the best decision based on this situation.

In practice, our ability to process information is limited. As a result, instead of taking into account all possible information about the object, we use a word (concept) to describe this notion, and then we make a decision based only on this word: e.g., a tiger or a dog. Instead of taking into account all the details of the fur and of the face, we decide to run away (if it is a tiger) or to wave in a friendly manner (if it is a dog).

In other words, instead of making an optimal decision for each object, we use the same decision based on an "average" object from the corresponding class. Since we make a decision without using all the information, based only on an approximate information, we thus lose some utility; see, e.g., [9] for a precise description of this loss.

From this viewpoint, the smaller the classes, the less utility we lose. This is what was used in the previous utility-based approaches to selecting basic level concepts.

However, if the classes are too small, we need to store and process too much information – and the need to waste resources (e.g., time) to process all this additional information also decreases utility. For example, instead of coming up with strategies corresponding to a few basic animals, we can develop separate strategies for short tigers, medium size tigers, larger tigers, etc. – but this would take more processing time and use memory resources which may be more useful for other tasks. While this is a concern, we should remember that we have billions of neurons, enough to store and process huge amounts of information, so this concern is rather secondary in comparison with a different between being eaten alive (if it is a tiger) or not (if it is a dog).

How to transform the above informal description of utility into precise formulas and how this leads to the desired explanations. The main reason for *disutility* (loss of utility) is that in a situation when we actually have an x, we use an approach which is optimal for a similar (but slightly different) object y. For example, instead of making a decision based on observing a very specific dog x, we ignore all the specifics of this dog, and we make a decision based only one the fact that x is a dog, i.e., in effect, we make a decision based on a "typical" dog y.

The larger the distance $d(x, y)$ between the objects x and y, the larger this disutility U. Intuitively, different objects within the corresponding class are similar to each other – otherwise they would not be classified into the same class. Thus, the distance $d(x, y)$ between objects from the same class are small. We can therefore expand the dependence of U on $d(x, y)$ in Taylor series and keep only the first few terms in this dependence. In general, $U = a_0 + a_1 \cdot d + a_2 \cdot d^2 + \ldots$ When the distance is 0, i.e., when $x = y$, there is no disutility, so $U = 0$. Thus, $a_0 = 0$ and the first non-zero term in the Taylor expansion is $U \approx a_1 \cdot d(x, y)$.

Once we act based on the class label ("concept"), we only know that an object belongs to the class, we do not know the exact object within the class. We may have different objects from this class with different probabilities. By the above property of utility, the resulting disutility of selecting a class is equal to the *average* value of the disutility – and is, thus proportional to the *average distance* $d(x, y)$ between objects from a given class. *This explains why average distance works better then the worst-case distance.*

When we go from a more abstract concept (i.e., from a larger class) to a more specific concept (i.e., to a smaller class of objects), the average distance decreases – and thus, the main part U_m of disutility decreases: $U'_m < U_m$. However, as we have mentioned, in addition to this main part of disutility U_m, there is also an additional secondary (smaller) part of utility $U_s \ll U_m$, which increases when we go to a more specific concept: $U'_s > U_s$.

On the qualitative level, this means the following: if the less general level has a much smaller degree of similarity (i.e., a drastically smaller average distance between the objects on this level), then selecting a concept on this less general level drastically decreases the disutility $U'_m \ll U_m$, and this decrease $U_m - U'_m \gg 0$ overwhelms the (inevitable) increase $U'_s - U_s$ in the secondary part of disutility, so that $U' = U_m + U'_s < U_m + U_s = U$. On the other hand, if the decrease in degree of similarity is small (i.e., $U'_m \approx U_m$), the increase in the secondary part of disutility $U'_s - U_s$ can over-stage the small decrease $U'_m - U_m$.

A basic level concept is a concept for which disutility U' is smaller than for a more general concept U and than for a more specific concept U''. In view of the above, this means that there should be a drastic difference between the degree of similarity U'_m at this level and the degree of similarity U_m at the more general level – otherwise, on the current level, we would not have smaller disutility. Similarly, there should be a small difference between the degree of similarity at the current level U'_m and the degree of similarity U''_m at the more specific

level – otherwise, on the current level, we would not have smaller disutility. *This explains the similarity approach in utility terms.*

Acknowledgments. We acknowledge support by the Operational Program Education for Competitiveness Project No. CZ.1.07/2.3.00/20.0060 co-financed by the European Social Fund and Czech Ministry of Education; it was partly performed when Martin Trnecka was visiting University of Texas at El Paso.

References

1. R. Belohlavek and M. Trnecka, "Basic level of concepts in formal concept analysis", In: *Proceedings of ICFCA'12*, Springer Verlag, Berlin, Heidelberg, 2012, pp. 28–44.
2. R. Belohlavek and M. Trnecka, "Basic level in formal concept analysis: interesting concepts and psychological ramifications", *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence IJCAI'2013*, Beijing, China, 2013, pp. 1233–1239.
3. J. E. Corter and M. A. Gluck, "Explaining basic categories: Feature predictability and information", *Psychological Bulletin*, 1992, Vol. 111, No. 2, pp. 291–303.
4. Fishburn P C. *Utility Theory for Decision Making*, John Wiley & Sons Inc., New York, 1969.
5. Fishburn P C. *Nonlinear Preference and Utility Theory*, The John Hopkins Press, Baltimore, Maryland, 1988.
6. D. H. Fisher, "Knowledge acquisition via incremental conceptual clustering", *Machine Learning*, 1987, Vol. 2, No. 2, pp. 139–172.
7. F. Gosselin and P. G. Schyns, "Why do we SLIP to the basic level? Computational constraints and their implementation", *Psychological Review*, 2001, Vol. 108, No. 4, pp. 735–758.
8. Keeney R L and Raiffa H. *Decisions with Multiple Objectives*, John Wiley and Sons, New York, 1976.
9. J. Lorkowski and V. Kreinovich, "Likert-Scale Fuzzy Uncertainty from a Traditional Decision Making Viewpoint: It Incorporates Both Subjective Probabilities and Utility Information", *Proceedings of the Joint World Congress of the International Fuzzy Systems Association and Annual Conference of the North American Fuzzy Information Processing Society IFSA/NAFIPS'2013*, Edmonton, Canada, June 24–28, 2013, pp. 525–530.
10. Luce R D and Raiffa R. *Games and Decisions: Introduction and Critical Survey*, Dover, New York, 1989.
11. G. L. Murphy, *The Big Book of Concepts*, MIT Press, Cambridge, Massachusetts, 2002.
12. Raiffa H. *Decision Analysis*, Addison-Wesley, Reading, Massachusetts, 1970.
13. E. Rosch, "Principles of categorization", In: E. Rosch and B. B. Lloyd, editors, *Cognition and Categorization*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1978, pp. 27–48.
14. M. D. Zeigenfuse and M. D. Lee, "A comparison of three measures of the association between a feature and a concept", In: L. Carlson, C. Holscher, and T. F. Shipley, editors, *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, Austin, Texas, 2011, pp. 243–248.

Towards a Physically Meaningful Definition of Computable Discontinuous and Multi-Valued Functions (Constraints)

Martine Ceberio, Olga Kosheleva, and Vladik Kreinovich

University of Texas at El Paso, El Paso, TX 79968, USA
{mceberio,olgak,vladik}@utep.edu

Abstract. In computable mathematics, there are known definitions of computable numbers, computable metric spaces, computable compact sets, and computable functions. A traditional definition of a computable function, however, covers only continuous functions. In many applications (e.g., in phase transitions), physical phenomena are described by discontinuous or multi-valued functions (a.k.a. constraints). In this paper, we provide a physics-motivated definition of computable discontinuous and multi-valued functions, and we analyze properties of this definition.

1 Formulation of the Problem

Need to define computable discontinuous functions. One of the main objectives of physics it to predict physical phenomena, i.e., use the observations to compute the predicted values of the corresponding physical quantities. Many physical phenomena such as phase transitions and quantum transitions include discontinuous dependencies $y = f(x)$ ("jumps"); see, e.g., [2].

In other physical situations, for some values x, we may have several possible values y. From the purely mathematical viewpoint, this means that the relation between x and y is no longer a function, it is a *relation* of a *constraint* $R \subseteq X \times Y$; following the terminology widely used in applications, we will also call them *multi-valued functions*.

To analyze which models of discontinuous or multi-valued behavior are computable and which are not, we need to have a precise definition of what is means for a discontinuous and/or multi-valued function to be computable. Alas, the current definitions of computable functions are mostly limited to continuous case.

What we plan to do. Our main goal is to define what it means for a discontinuous and/or multi-valued function to be computable.

For that purpose, we first explain the current definitions of computable numbers, objects, and functions. Then, we use physical motivations to come up with a new definition of computable discontinuous and multi-valued functions. Finally, we provide a few preliminary results about the new definition.

Computable numbers: reminder. Intuitively, a real number is *computable* if we can compute it with any desired accuracy. In more precise terms, a real number x is called *computable* if there exists an algorithm that, given a natural number n, returns a rational number r_n which is 2^{-n}-close to x: $|x - r_n| \le 2^{-n}$; [1, 3].

Computable metric spaces: motivation. A similar notion of computable elements can be defined for general metric spaces. In general, a element x is computable if there is an algorithm which generates better and better approximation to x. At each moment of time, we only have a finite amount of information about x; based on this information, we produce an approximation corresponding to this information. Any information can be represented, in the computer, as a sequence of 0s and 1s; any such sequence can be, in turn, interpreted as a binary integer n. Let $\widetilde{x}_n$ denote an approximation corresponding to an integer n. Then, it makes sense to require that in a computable metric space, there is a sequence of such approximating elements $\{\widetilde{x}_n\}$.

Computable means, in particular, that the distance $d_X(\widetilde{x}_n, \widetilde{x}_m)$ between such elements should be computable. Thus, we arrive at the following definition.

Computable metric spaces: definition. By a computable metric space, we mean a metric space X with a sequence $\{\widetilde{x}_n\}$ of elements such that there is an algorithm that, given two natural numbers m and n, returns the distance $d_X(\widetilde{x}_m, \widetilde{x}_n)$ (i.e., for every natural number k, returns a rational number r_k which is 2^{-k}-close to $d_X(\widetilde{x}_m, \widetilde{x}_n)$).

We say that an element x of a computable metric space X is *computable* if there exists an algorithm that, given a natural number n, returns an integer k_n for which $\widetilde{x}_{k_n}$ is 2^{-n}-close to x: $d_X(\widetilde{x}_{k_n}, x) \le 2^{-n}$.

Computable functions: definition. A function $f : X \to Y$ from a computable metric space X to a computable metric space Y is called *computable* if there exists an algorithm which uses x as an input and computes, for each integer n, a 2^{-n}-approximation y_k to $f(x)$. By "uses x as an input", we mean that to compute y_k, this algorithm can request, for each integer m, a 2^{-m}-approximation x_ℓ to x (and to use the index ℓ of this 2^{-m}-approximation in computing y_k).

Computable functions are continuous. The problem with the above definition is that all the functions computable according to this definition are continuous; see, e.g., [1, 3]. Thus, we cannot use this definition to check how well we can compute a discontinuous function.

This continuity is easy to understand. For example, if we have a function $f(x)$ form real numbers to real numbers which is equal to 0 for $x \le 0$ an to 1 for $x > 0$, then, if we could compute $f(x)$ for a given x with accuracy 2^{-2}, then we would be able, given a computable real number x, to tell whether this number is positive or not, and this is known to be algorithmically impossible.

Computable compact set. In analyzing computability, it is often useful to start with *pre-compact* metric spaces, i.e., metric spaces X for which, for every positive real number $\varepsilon > 0$, there exists a finite *ε-net*, i.e., a finite list of elements L such that every element $x \in X$ is ε-close to one of the elements from this list. In a Euclidean space, every bounded set is compact. A pre-compact set is *compact* if every converging sequence has a limit.

A natural idea is to call a compact metric space X *computable compact* if X is a computable metric space and there is an (additional) algorithm that, given an integer n, returns a finite list L_n of elements of X which is a 2^{-n}-net for X.

2 Towards A New Definition of Computable Discontinuous and Multi-Valued Functions

Simplifying comment. Before we start analyzing the problem, let us make one important comment. Functions can not only be discontinuous or multi-valued, they can also be undefined for some inputs x. However, in contrast to discontinuity and multiplicity of values, this is not a serious problem: if a relation is not everywhere defined, we can make it everywhere defined if we consider, instead of the original set X, a projection of R on this set. For example, a function $\sqrt{x}$ is not everywhere defined on the real line, but it is everywhere defined on the set of all non-negative real numbers. Thus, without losing generality, we can assume that our relation is everywhere defined.

Definition 1. *A relation $R \subseteq X \times Y$ is called* everywhere defined *if for every $x \in X$, there exists a $y \in Y$ for which $(x, y) \in R$.*

Analysis of the problem. From the physical viewpoint, what does it mean that the dependence between x and y – as described by a given discontinuous and/or multi-valued function – is computable?

In the ideal case, when we have a continuous single-valued dependence, the value x uniquely determines the value $y = f(x)$. In this case, once we know x, we want to compute $f(x)$ with a given accuracy. This is exactly the idea behind the usual definition of a computable function.

For a multi-valued function, for the same input x, we may get several different values y. In this case, it is desirable to compute the *set* of all possible value y corresponding to a given x. When we limit ourselves to multi-valued mappings from a compact set X to a compact set Y, the set of x-possible values of y is pre-compact, and thus, with any given accuracy, can be described by a finite list L of possible values. In other words:

- first, the list L should represent *all* possible values, i.e., if y is a possible value of $f(x)$ for a given x, then y should be close to one of the values from the finite list L;
- second, all the values from the list L must be possible values; in other words, for every value from the list, there must exist a close possible value of $f(x)$.

Discontinuity provides an additional complexity which can be illustrated on the example of the above discontinuous function $f(x) = 0$ for $x \le 0$ and $f(x) = 1$ for $x > 0$. In particular, for $x = 0$, we get $f(x) = f(0) = 0$. However, at each stage of the computation, we only know an approximate value of x. So, when the actual value of the input is $x = 0$, we will never find out whether x is non-positive (in which case $f(x) = 0$) or positive (in which case $f(x) = 1$). Thus, no matter how accurately we measure x, the only information about y that we can conclude

is y is either equal to 0 or equal to 1. In general, we need to take into account not only the values $f(x)$ for a given x, but also the values $f(x')$ corresponding to values x' which are close to x. In view of this, the above properties of the list L must be appropriately modified:

- first, the list L should represent *all* possible values, i.e., if y is a possible value of $f(x')$ for some x' which is close to the given x, then y should be close to one of the values from the finite list L;
- second, all the values from the list L must be possible values; in other words, for every value from the list, there must exist a close value y which is a possible value of $f(x')$ for some x' which is close to x.

In general, the closeness does does not have to be the same in both cases. Thus, we arrive at the following definition.

Definition 2. *Let X and Y be computable compact sets with metrics d_X and d_Y. An everywhere defined relation $R \subseteq X \times Y$ is called* computable *if there exists an algorithm that, given a computable element $x \in X$ and computable positive numbers $0 < \varepsilon < \varepsilon'$ and $0 < \delta$, produces a finite list $\{y_1, \ldots, y_m\} \subseteq Y$ that satisfies the following two properties:*

(1) *if $(x', y) \in R$ for some x' for which $d_X(x', x) \le \varepsilon$, then there exists an i for which $d_Y(y, y_i) \le \delta$;*
(2) *for each element y_i from this list, there exist values x' and y for which $d_X(x, x') \le \varepsilon'$, $d_Y(y_i, y) \le \delta$, and $(x', y) \in R$.*

3 Properties of the New Definition

Main result. If X and Y are metric spaces with metrics d_X and d_Y, then on their Cartesian product $X \times Y$ (i.e., the set of all pairs (x, y), $x \in X$ and $y \in Y$) we can define a metric $d_{X\times Y}((x, y), (x', y')) \stackrel{\text{def}}{=} \max(d_X(x, x'), d_Y(y, y'))$. One can check that if X and Y are both compact sets, then the product $X \times Y$ is also a compact set: to get an ε-net for $X \times Y$, it is sufficient to take ε-nets L_X for X and L_Y for Y; one can then easily check that the set $L_X \times L_y$ of all possible pairs is an ε-net for the Cartesian product $X \times Y$. This construction is computable, so we conclude that the Cartesian product of computable compact sets is also a computable compact set.

Our first – somewhat surprising – result is that this new definition is equivalent to simply requiring that the set R (describing the graph of the relation) is a computable compact set:

Proposition 1. *Let X and Y be computable compact sets. A relation $R \subseteq X \times Y$ is computable if and only if the set R is a computable compact set.*

Proof. $\Leftarrow$ Let us first prove that if R is a computable compact set, then the relation R is computable in the sense of Definition 2. Indeed, let x be a computable element of X, and let the computable positive values $\varepsilon < \varepsilon'$ be given. Then, according to a known result from [1], we can find a computable value $\varepsilon_0 \in (\varepsilon, \varepsilon')$ for

which the set $S \stackrel{\text{def}}{=} \{(x', y) \in R : d_X(x, x') \le \varepsilon_0\}$ is also a computable compact set. Thus, for a given computable number $\delta > 0$, there exists a finite δ-net for this set S. Let us denote the elements of this δ-net L by $(x_1, y_1), \ldots, (x_m, y_m)$. Let us show that, as the desired finite list, we can now take the list $\{y_1, \ldots, y_m\}$. Let us prove that this list satisfies both desired properties.

(1) If $(x', y) \in R$ for some x' for which $d_X(x, x') \le \varepsilon$, then, due to $\varepsilon < \varepsilon_0$, we have $d_X(x, x') < \varepsilon_0$. Thus, $(x', y) \in S$. Since $L = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ is a δ-net for the set S, we conclude that there exists an index i for which $d_{X\times Y}((x', y), (x_i, y_i)) \le \delta$. By definition of $d_{X\times Y}$, this means that $\max(d_X(x', x_i), d_Y(y, y_i)) \le \delta$ and therefore, $d_Y(y, y_i) \le \delta$. The first property from Definition 1 is proven.

(2) Let us now prove the second property. Let y_i be one of the selected elements. By our construction, the corresponding pair (x_i, y_i) belongs to δ-net for the set S. In particular, this means that $(x_i, y_i) \in S$. This means that $(x_i, y_i) \in R$ and that $d_X(x, x_i) \le \varepsilon_0$. Since $\varepsilon_0 < \varepsilon'$, we conclude that $d_X(x, x_i) \le \varepsilon'$. Thus, for each i, there exists $x' = x_i$ and $y = y_i$ for which $d_X(x, x') \le \varepsilon'$, $d_Y(y_i, y) = 0 \le \delta$, and $(x', y) \in R$. The second property is proven as well.

$\Rightarrow$ Let us now prove that if R is a computable relation in the sense of Definition 2, then R is computable compact set. For that, we must show how, given a computable positive real number $\alpha > 0$, we can generate an α-net for this set R. First, we use that fact that X is a computable compact, and generate an $(\alpha/2)$-net $\{x_1, \ldots, x_k\}$. For each point x_i, we then apply Definition 2 for $\delta = \varepsilon = \alpha/2$ and $\varepsilon' = \alpha$ and generate the corresponding list $\{y_{i1}, \ldots, y_{im_i}\}$. Let us show that the pairs (x_i, y_{ij}) form an α-net for the set R.

Indeed, by Definition 2, for each i and j, there exist values x' and y for which $d_X(x_i, x') \le \varepsilon' = \alpha$, $d_Y(y_{ij}, y) \le \delta = \alpha/2$, and $(x', y) \in R$. Thus, the pair (x_i, y_{ij}) is α-close to an element $(x', y) \in R$.

Vice versa, let $(x, y) \in R$. Since x_i form an $(\alpha/2)$-net, there exists an i for which $d(x, x_i) \le \alpha/2 = \varepsilon$. From Property (1) of Definition 2, we can now conclude that there exists a j for which $d_Y(y, y_{ij}) \le \delta = \alpha$. Thus, $d_{X\times Y}((x, y), (x_i, y_{ij})) = \max(d_X(x, x_i), d_Y(y, y_{ij})) \le \max(\alpha/2, \alpha) = \alpha$. The proposition is proven.

Inverse functions: a corollary. If the range of R is the whole set Y, then, from Proposition 1, it follows that a multi-valued function (relation) R is computable if and only if its inverse $R^{-1} = \{(x, y) : (y, x) \in R\}$ is computable.

Acknowledgments. This work was supported in part by NSF grants HRD-1242122 and DUE-0926721, by NIH Grants 1 T36 GM078000-01 and 1R43TR000173-01, and by an ONR grant N62909-12-1-7039.

References

1. Bishop, E., Bridges, D. S.: Constructive Analysis, Springer, N.Y. (1985)
2. Feynman R., Leighton R., Sands M.: The Feynman Lectures on Physics, Addison Wesley, Boston, Massachusetts (2005)
3. Weihrauch, K.: Computable Analysis, Springer-Verlag, Berlin (2000)

Energy-efficient Automatic Memory Management for Delay-Intolerant Systems

Gabriel Arellano, Ed Dragone, Eric Freudenthal, Ed Hudgins, Md. Jahid, Roger Ochoa, David Pruitt, Adrian Veliz

Department of Computer Science
University of Texas at El Paso
El Paso Texas, United States
grarellano@miners.utep.edu, edragone@miners.utep.edu, efreudenthal@utep.edu, eghudgins@miners.utep.edu, kajalcse2001@gmail.com, rochoa9@miners.utep.edu, ddpruitt@miners.utep.edu, aeveliz@miners.utep.edu
http://robust.cs.utep.edu

Abstract. A significant fraction of mobile and infrastructure computing is implemented in garbage collected (GC'd) programming languages. For example, Java is the primary programming language for the more than 109 delay-intolerant and energy limited devices running Android. Java is also employed at online data centers, which are also delay intolerant and consume 10% of US electrical capacity. Garbage collection requires a periodic time and energy-intensive operation that can contribute to program execution delays. // // Techniques for reducing mutator (application program) delays and minimizing heap size for automatic memory management (AMM) have been investigated independently. However, there is a paucity of research related to the composition of these constraints, which is the focus of our work. // // We examine AMM policies related to heap size management and GC scheduling. Due to its market relevance the the availability of source code, our research target is Android's Dalvik (J)VM. Our instrumentation of recent Android releases indicates that it is not unusual for 20% of CPU time to be spent garbage collecting, and for mutator threads to be blocked during a significant portion of this time. Furthermore, prolonged low-priority garbage collection operations can trigger CPU clock frequency increases, and may result in a disportionate increase in energy consumption. We have not yet examined whether these techniques are suitable for reducing energy consumption by infrastructure computing.

Heap footprint size is inversely related to the frequency of memory exhaustion and GC. E.g., a small heap will be exhausted quite rapidly (and frequently trigger low-yield GC). Android's Dalvik automatic memory management policies for heap growth and garbage collection scheduling utilize constant factors tuned to minimize memory footprint. These policies result in only marginally acceptable response times and garbage collection significantly contributes to apps' CPU time and therefore energy consumption. Initial sensitivity studies indicate that the contribution of GC towards an Android app's total CPU time can be significantly reduced by permitting a moderate amount of heap growth.

Dalvik's AMM policy allows for a moderate amount of heap growth that is a constant fraction of free space at the time a GC completes. Should a heap be unable to satisfy a thread's memory allocation request, a GC is initiated and the thread's execution is blocked until its request can be satisfied. Additional heap growth is only permitted when needed to satisfy such threads' memory allocation requests after the GC completes. Our instrumentation of Dalvik indicates that the heap growth permitted by Dalvik's policies is insufficient to avoid frequent low-yield GCs. These frequent GC operations delay program execution for, increase CPU time consumed by, and trigger clock frequency increases for high value apps such as Google Maps.
Our alternative policies relax constraints on heap growth when memory exhaustion occurs frequently and dynamically determine when background garbage collection should be commenced based upon the amount of free heap space and the rate of program memory allocation.
The resulting system that has no significant delays due to garbage collection and only a moderate (and acceptable) increase in heap size.
Current efforts are focusing on

- (Determining how to) Characterize energy consumption caused by GC.
- Developing an experimental methodology and infrastructure for collecting and analyzing memory subsystem performance data from a large number of devices.
- Extending instrumentation to identify apps with bursty allocation patterns and developing policies that reduce memory size by scheduling GC after such bursts.
- Examining approaches to enable a frequency governor to identify GC threads and GC urgency when characterizing CPU load and setting CPU frequency. ...

Keywords: Garbage Collection, background, foreground, automatic memory management, power limited, delay intolerant, memory exhaustion

Predicting memory exhaustion by evaluating runtime features using machine learning

Edward Hudgins, Gabriel Arellano, Eduardo Dragone,
Eric Freudenthal, Olac Fuentes, M. D. Jahid,
Rogelio Ochoa, David Pruitt, and Adrian Veliz

University of Texas at El Paso, Department of Computer Science,
500 West University Avenue, El Paso, TX 79968
{eghudgins,grarellano,edragone,majahid,
rochoa9,ddpruitt,aeveliz}@miners.utep.edu
{efreudenthal,ofuentes}@utep.edu
https://sites.google.com/site/robustautonomicsystems/

Abstract. Runtime support for automatic memory management (AMM) requires automatic identification of recyclable memory. This process is commonly called garbage collection and abbreviated GC. This identification process is commonly implemented using a "tracing" garbage collector that implements a transitive search of reachable memory objects. AMM can interfere with program responsiveness: Should a runtime heap become exhausted, program execution may need to be blocked until garbage collection has freed an adequate amount of memory to permit its continuance.

In order to minimize GC delays, it is now common to proactively schedule garbage collection when the CPU would otherwise be idle. The frequency that GC is required is strongly dependent on the amount of memory available for dynamic allocation: A heap with a smaller memory "footprint" will require more frequent GC. Since most modern processors can be programmed to enter a low-power idle state when the ready queue is empty and frequency governors typically increase clock frequency when CPU utilization is high, small heap footprints or paranoid GC invocation policies can result in significant increases in energy consumption. Our research examines approaches for dynamically determining (1) the amount of memory that should be available for dynamic allocation (heap footprint size) and (2) times when the garbage collector's background execution should be invoked.

Factors that affect these decisions can include:

- The expected amount of time before the heap will be exhausted.
- The expected amount of time required to execute the garbage collector
- The expected fraction of idle CPU cycles available to execute the garbage collector without delaying the user program (commonly called the "mutator")
- The expected rate of heap exhaustion

Empirical measurement of the mobile systems we are targeting indicate that background execution of the garbage collector typically completes within 200ms, and that GC CPU time allocation is below 5% if executed no more frequently than once every 2s.

We have developed policies that permit heap footprint growth to prevent GC from occurring more frequently than once every 2s and have found that the resulting increase of heap size (by as much as three times) never exceeded 16MB, and constituted a small fraction of total system RAM and therefore was acceptable. Our focus has been on the complementary problem of predicting the time of heap exhaustion so that GC can be initiated just-in-time to complete prior to heap exhaustion.

We are exploring both human and machine-designed heuristics. This paper describes a machine learning (ML) approach being evaluated for machine-designed heuristics. As is common for ML, humans still play a role in choosing the family of ML classifier and in designing the attributes available to the ML system. Operational constraints have driven our initial selection of candidate classifiers and features. The remainder of this abstract describes these constraints, the candidate classifiers and features being examined, experimental approach, and early experimental results. An online classification algorithm responsible for invoking GC will be integrated into the runtime system and must consume little CPU time. This online decision process is distinct from offline training, for which prolonged computation is acceptable.

Offline training uses data collected from a mobile device instrumented to log the sequence of significant memory allocation events. In order to understand the behavior that precedes memory exhaustion, background garbage collection is disabled, and instead a foreground GC is invoked when the heap is unable to satisfy a memory request. If offline analysis indicates superior characteristics to human-designed, we will pursue embedding the classifier into an operational system.

Our initial candidate classifiers are decision-trees generated by Ross Quinlan's C 4.5 algorithm. Since these decision trees do not perform arithmetic computations, we have found it beneficial to synthesize features that directly correspond to memory allocation rates, availability, and their ratios. Their ratio provide a first-order approximation to the expected time to memory exhaustion.

Keywords: Garbage collection, background, foreground, machine learning, C4.5, automatic memory management, power-limited, delay-intolerant, memory exhaustion

Prioritizing Weak Reference Garbage Collection to Facilitate Object Caching

Adrian Veliz and Eric Freudenthal

University of Texas at El Paso, Computer Science,
500 W. University Ave., 79968 El Paso, Texas, USA
adrian.veliz@miners.utep.edu,eric.freudenthal@utep.edu
http://robust.cs.utep.edu

Abstract. Caching of computed or fetched values for potential future reuse is a common programming idiom. Ideally, management policies should be driven by the availability of storage resources and expected future program behavior. A cache's retention policy can affect memory footprint size, program responsiveness, energy consumption, and input/output operations.

Ideally retention policies will be sensitive to the availability of (memory) storage resources and (predicted) future reference sequences. Such policies would require cooperative interaction among programs and the memory allocation (and reclamation) system. This cooperation is impeded by traditional APIs for memory allocation that do not expose resource availability.

Weak references are a language construct for garbage collected systems intended to permit programs to inform the memory management system of objects whose eviction can be tolerated. A weak reference does not prevent an object from being evicted by the garbage collector. Weak references to recycled objects are automatically cleared, and interfaces are provided that inform programs when a weakly referenced object is evicted.

Weak references are a mature linguistic construct. Java, Haskell, SML, Python, and C# implement weak references with semantics intended to permit this cooperation between memory management systems and programs that can tolerate the eviction of referenced reachable objects. Recently, the term weak reference has been overloaded by the designers of the version of Objective C used to implement apps for Apple IOS devices in a manner that enables its reference-count garbage collector to break cycles and therefore evict unreachable objects. This alternative implementation is incompatible with the autonomic cache designs we are advocating.

While the linguistic semantics for weak references are mature, there is a paucity of research examining retention strategies that are sensitive to program behavior. A typical policy will clear all weak references upon heap exhaustion. Our objective is to examine whether an object's future value can be effectively prioritized by the memory management system

and how such prioritization might be used to guide retention policies for weakly-referenced objects. Potential value attributes include (1) probability of future access, (2) cost of object reconstruction (if needed), and (3) the amount of memory that will be reclaimed should the object be recycled.

Like trace-driven memory cache analysis techniques, our research will collect reference histories to weakly referenced objects on a system modified to not recycle weak references. Defensive programming within common apps may distort these results. It would be useful to team up with app developers who could disable the defensive components in versions used for this evaluation.

The replacement cost for an evicted object include interface delays incurred due to computation or network transfers and energy consumption by the CPU and network subsystems. We are examining whether instrumentation of an object's constructor can yield data useful to guide retention policies. An alternative approach would be to provide an API that permits programs to specify metrics relevant for predicting replacement cost.

Finally, we are also examining whether weak referents should be evicted on-demand, or instead proactively prior to heap exhaustion. Eviction decisions require computation, and eviction protocols for weakly referenced objects may involve the execution of finalizers and notification of apps that implement listeners for such events. Since proactive evictions occur prior to heap exhaustion, these computations can avoid delaying the execution of mutator threads.

Keywords: Weak References, Garbage Collection, Cache Eviction

Exploring the Effects of Emotive Stimuli in Presidential Policy Appeals

José D. Villalobos and Cigdem V. Sirin

University of Texas at El Paso

Abstract. The proposed research applies a "thematic relevance of emotions" framework to examine public reactions to presidential policy appeals. To date, no research has explored whether the relevance of emotive stimuli utilized by presidents in their public messages produces differential effects on public opinion. We investigate whether the thematic relevance of emotive stimuli embedded in presidential speeches influences people's risk perceptions and policy support in the context of military interventions in civil conflict. This study applies the "social risk amplification" framework to further scholarly understanding of public reactions to presidential policy appeals. We investigate whether and to what extent the thematic relevance of emotive stimuli embedded in presidential speeches influences people's risk perceptions and preferences regarding military interventions in civil conflict. In our initial pilot study, we find that while the induction of anger via thematically relevant emotive triggers did not lead to lower risk perceptions, it did lead to higher levels of support regarding military action. By comparison, we find that the effects of anger on policy choice observed in the thematically irrelevant emotive trigger condition are not significantly different from the emotion-neutral control condition. Moving forward, we plan to conduct a survey experiment using a nationally representative random sample. Our study offers a new contribution to the literature on risk perceptions, risk communication, public support for military interventions, and executive power and accountability in the public presidency.

1 Introduction

In recent years, a growing body of research has set out to examine the role that emotions play in shaping political attitudes and behavior (e.g., [2, 12]). Therein, several studies have begun exploring the distinct effects of different emotions on people's risk perceptions and political judgments regarding various issues, events, and choices (see, for example, [7, 11]). Overall, this line of research has found that anger is likely to increase one's propensity to favor retaliatory, risky, and aggressive policy actions compared to other emotions such as fear and sadness.

On par with such work, research on political discourse and communication has shown that emotion-laden appeals are selectively deployed by political leaders in order to (1) reboost their public approval ratings while suppressing criticism and dissent, (2) justify and garner support for their policy agenda (including

military intervention policies), and (3) divert the attention of the public during times of domestic turmoil (see, for example, [3]). Within the American context, presidents often resort to emotive appeals in wielding the bully pulpit with the expectation that it may help rally public support behind their political and policy agendas. For example, a president might seek empathy from listeners by describing how civilians in a foreign country may be suffering at the hands of an oppressive, violent regime and asking Americans to support a plan for military action. To what extent might these emotive appeals influence the public mood and lend credence to a president's policy preferences, both with respect to a particular policy (e.g., military intervention in civil conflict) and a president's broader policy agenda (e.g., attempting to take advantage of such emotion-driven public rallies to seek out other foreign and domestic policy objectives)?

Despite heightened scholarly interest concerning emotion-laden appeals and public reactions, one major issue that merits further investigation is the potential link between the antecedent conditions that trigger a particular emotion and the effects that such emotion has on one's decision making (see [13]). Is it of importance whether the specificcause of a certain emotional state is thematically linked to the decision task at hand? For instance, does angerregardless of its sourcetend to lower people's perceptions of risk and increase support for an aggressive foreign policy option or do the thematic underpinnings of anger (i.e., the specific contents that trigger such emotion, such as an emotional story embedded in a presidential speech concerning a civil conflict abroad) matter vis--vis the policy choice? To address these questions, this study examines whether and to what extent the thematic relevance of emotive stimuli embedded in presidential speeches influences people's risk perceptions and support regarding military interventions in civil conflict. Therein, we consider two alternative theoretical perspectives: (1) the functional autonomy of emotions versus (2) the thematic relevance of emotions.

Applying the former perspective, one may argue that the effects of emotions are functionally autonomous from their sources such that a particular emotional state (such as being angry) will have a uniform effect on one's decision making irrespective of the thematic content of the emotive trigger as its source (e.g., [4, 18]). According to this view, appraisal tendencies generated by specific emotions can persist and spill over to influence one's political judgment on a given issue even when the target of judgment is unrelated to the emotion-eliciting stimulus ([4, 8]).

Alternatively, it is also plausible to argue that the effects of a certain emotional state on one's decision making is conditional on whether the specific source of that emotional state is thematically related to the policy issue at hand. [18] suggest that although people have a tendency to attribute their affective states to the current object of attention (even when the actual source of their feelings is completely unrelated to that object), such misattributions typically disappear when people become aware of the true source of their affective states (see also [10]). Furthermore, [15] point out that intense emotions are likely to be particularly resistant to misattribution since their sources tend to be highly salient.

Accordingly, the extent to which an emotional state influences subsequent political judgments and assessments of risk may be contingent on whether the individual perceives an external stimulus as connected to and/or responsible for such emotional state ([10, 14]). Within such context, an individual may consider a given emotion to be an irrelevant source of information if the decision making domain under evaluation is unrelated to the emotional state.

Of these two contending theoretical perspectives, we expect that the thematic relevance of emotions (rather than the functional autonomy of emotions) is more likely to be the primary underlying mechanism regarding the effects that emotions have on public reactions to presidential messages. Specifically, we posit that because an emotive trigger is embedded in a given cognitive context, the relevance of that context to a policy issue under consideration is likely to accentuate (or diminish) the impact of the induced emotional state on an individual's risk perceptions and decision making.

In the context of military interventions, if the source of a certain emotional state is also thematically relevant to the issue (such as getting angry after exposure to an emotional story embedded in a presidential speech addressing a civil conflict abroad), we believe that the salience of such source is likely to amplify the impact of the elicited emotion on people's risk perceptions and policy preferences. By comparison, if the source of a certain emotional state is not thematically relevant to the issue (such as getting angry after exposure to a presidential speech addressing an incident involving crime in U.S. cities), the disconnect between the source of the emotional state (crime in U.S. cities) and the policy issue to be considered (military intervention in civil conflict abroad) may subdue the impact of the elicited emotion on people's policy preferences. As such, our arguments regarding the effects of thematic relevance of emotions on reactions to civil conflict parallel the "social amplification of risk" framework, which suggests that an adverse event interacts with various psychological, social, institutional, and cultural processes that may subsequently amplify (or attenuate) people's responses to the event (see [1, 9, 22]). Consequently, such interaction triggers risk-related behavior.

2 Theoretical Framework and Hypotheses

In this study, our focus is on examining the effects of anger induced by thematically relevant and irrelevant emotive presidential appeals on people's risk perceptions and policy preferences regarding military intervention in civil conflict. Therein, we consider two alternative theoretical perspectives: (1) the functional autonomy of emotions versus (2) the thematic relevance of emotions.

Applying the former perspective, one may argue that the effects of emotions are functionally autonomous from their sources such that a particular emotional state (such as being angry) will have a uniform effect on one's decision making irrespective of the thematic content of the emotive trigger as its source (e.g., [4, 5, 11, 18, 17]). According to this view, appraisal tendencies generated by specific emotions can persist and spill over to influence one's political judgment on a

given issue even when the target of judgment is unrelated to the emotion-eliciting stimulus ([4, 5, 8]).

Alternatively, it is also plausible to argue that the effects of a certain emotional state on one's decision making is conditional on whether the specific source of that emotional state is thematically related to the policy issue at hand. [18, 17] suggest that although people have a tendency to attribute their affective states to the current object of attention (even when the actual source of their feelings is completely unrelated to that object), such misattributions typically disappear when people become aware of the true source of their affective states (see also [10, 19, 20]). Furthermore, [15] point out that intense emotions are likely to be particularly resistant to misattribution since their sources tend to be highly salient. Accordingly, the extent to which an emotional state influences subsequent political judgments and assessments of risk may be contingent on whether the individual perceives an external stimulus as connected to and/or responsible for such emotional state [6, 10, 14]. Within such context, an individual may consider a given emotion to be an irrelevant source of information if the decision making domain under evaluation is unrelated to the emotional state.

These two alternative theoretical perspectives concerning the effects of thematically relevant, irrelevant, and neutral emotive triggers may be expressed as follows (where "E" denotes emotive trigger, "R" denotes thematic relevance, "IR" denotes thematic irrelevance, and "C" denotes control):

1) $[E_R \approx E_{IR}] \neq C$ (functional autonomy of emotions)
2) $E_R \neq [E_{IR} \approx C]$ (thematic relevance of emotions)

Of these two contending theoretical perspectives, we expect that the thematic relevance of emotions (rather than the functional autonomy of emotions) is more likely to be the primary underlying mechanism regarding the effects that emotions have on public reactions to presidential messages. Specifically, we posit that because an emotive trigger is embedded in a given cognitive context, the relevance of that context to a policy issue under consideration is likely to accentuate (or diminish) the impact of the induced emotional state on an individual's risk perceptions and decision making.

In the context of military interventions, if the source of a certain emotional state is also thematically relevant to the issue (such as getting angry after exposure to an emotional story embedded in a presidential speech addressing a civil conflict abroad), we believe that the salience of such source is likely to amplify the impact of the elicited emotion on people's risk perceptions and policy preferences. By comparison, if the source of a certain emotional state is not thematically relevant to the issue (such as getting angry after exposure to a presidential speech addressing an incident involving crime in U.S. cities), the disconnect between the source of the emotional state (crime in U.S. cities) and the policy issue to be considered (military intervention in civil conflict abroad) may subdue the impact of the elicited emotion on people's policy preferences. As such, our arguments regarding the effects of thematic relevance of emotions on reactions to civil conflict parallel the "social amplification of risk" framework, which suggests that an adverse event interacts with various psychological, social,

institutional, and cultural processes that may subsequently amplify (or attenuate) people's responses to the event (see [1, 9, 22]). As a result of such interaction, one may expect a higher tendency to engage in risk-related behavior and support for forceful policy options to deal with an adverse event (particularly if anger is the dominant emotion aroused in response). Accordingly, we hypothesize as follows:

> Hypothesis 1: *Emotive presidential appeals for taking military action that include thematically relevant triggers of anger are likely to prompt lower risk perceptions among individuals (i.e., as compared to presidential appeals that include thematically irrelevant triggers of anger or appeals that do not include any emotive triggers).*
>
> Hypothesis 2: *Emotive presidential appeals for taking military action that include thematically relevant triggers of anger are likely to prompt higher support among individuals (i.e., as compared to presidential appeals that include thematically irrelevant triggers of anger or appeals that do not include any emotive triggers).*

3 Experimental Design

To test our hypotheses, we conducted an experiment with a total of two-hundred-twenty-one undergraduate students. Our experiment involves a between-groups factorial design consisting of three conditions: (1) thematically relevant emotive trigger condition, (2) thematically irrelevant emotive trigger condition, and (3) emotion-neutral control condition. We randomly assigned the participants to the experimental conditions.

For the manipulation of experimental factors, we designed hypothetical presidential speeches. Specifically, participants read a short presidential speech addressing two separate issues: (1) the issue of crime in U.S. cities and (2) the possibility of a military intervention in a hypothetical civil conflict abroad. In the thematically relevant emotive trigger condition, an emotional, anger-inducing story was embedded in the speech as part of the civil conflict issue. In contrast, in the thematically irrelevant emotive trigger condition, the equivalent emotional story was embedded in the speech as part of the issue concerning crime in U.S. cities. No emotional story was provided in the emotion-neutral control condition. To obtain maximum internal validity and control, all the wording used in the scenarios across the different experimental conditions were kept constant except for the induction and placement of the emotional story.

After their exposure to the presidential speech, participants were asked about their risk perceptions and support concerning a possible military intervention in said civil conflict. The response options for the question regarding the level of perceived risk were coded as "1" for "not at all risky," "2" for "Not too risky," "3" for "Somewhat risky," "4" for "Fairly risky," and "5" for "Very risky." As for the question regarding the level of support, we assigned a code of "1" for "Not support at all," "2" for "Not much support," "3" for Somewhat support," "4" for "Fairly support," and "5" for "Strongly support."

To ensure that our experimental design is internally valid, we conducted several manipulation checks. Specifically, we asked the participants how angry they felt about the information they were exposed to. The response options ranged from "1" for "Not at all angry" to "5" for "Very angry." The results show that participants in both thematically relevant and irrelevant emotive trigger conditions expressed significantly higher levels of anger compared to emotion-neutral control condition ($p < .001$). This result indicates that the emotive trigger embedded in the presidential speech was indeed effective in inducing anger. Furthermore, there was no statistically significant difference regarding the level of anger induced between the thematically relevant and irrelevant emotive trigger conditions ($p > .10$). This result suggests that the experimental effects observed upon exposure to emotive triggers are mainly due to the thematic relevance of such triggers as we intended to manipulate and not due to other confounding factors such as the variations in the intensity of the triggered emotional state.

Last, we also asked the participants whether they experience several other emotions, including sadness, fear, worry, and anxiety in reaction to the information they read. This is because individuals can experience several emotions at any given moment and one emotion may nullify or mediate the impact of another. The results demonstrate that the reported levels of fear, worry, and anxiety are not significantly different than the emotion-neutral control condition ($p > .10$). Sadness is statistically significant, though the mean level of anger (M_R = X and M_{IR} = X) is higher than the mean level of sadness (M_R = X and M_{IR} = X) expressed by the participants in both the thematically relevant and irrelevant conditions[1]. The results thus suggest that anger is the dominant emotion induced by the experimental stimuli. In all, these manipulation checks confirm the internal validity of our experiment.

References

1. Burns, W. J. and P. Slovic. 2007. "The Diffusion of Fear: Modeling Community Response to a Terrorist Strike. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology 4(4): 298–317.
2. Conover, P. J., and S. Feldman. 1986. "Emotional Reactions to the Economy: Im Mad as Hell and Im not Going to Take it Anymore." American Journal of Political Science 30(1): 30–78.
3. De Castella, K., C. McGarty, and L. Musgrove. 2009. "Fear Arousing Content in Political Rhetoric About Terrorism: An Analysis of Speeches by Australian Prime Minister Howard." Political Psychology 30(1): 1–26.
4. Gasper, K. and G. L. Clore. 1998. "The Persistent Use of Negative Affect by Anxious Individuals to Estimate Risk." Journal of Personality and Social Psychology 74(5): 1350–1363.

[1] It is also important to note that even if other emotions are intervening with the predicted effects of anger, our experimental results would be more, not less, conservative because it would demonstrate that the hypothesized effects for anger still hold even in the presence of other emotional states that would otherwise nullify or mediate the impact of anger.

5. Goldberg, J. H., J. S. Lerner, and P. E. Tetlock. 1999. “Rage and Reason: The Psychology of the Intuitive Prosecutor.” European Journal of Social Psychology 29(5–6): 781–795.
6. Hirt, E. R., G. M. Levine, H. E. McDonald, and R. J. Melton. 1997. “The Role of Mood in Quantitative and Qualitative Aspects of Performance: Single or Multiple Mechanisms?” Journal of Experimental Social Psychology 33(6): 602–629.
7. Huddy, L., S. Feldman, C. Taber, and G. Lahav. 2005. “Threat, Anxiety, and Support of Anti–Terrorism Policies.” American Journal of Political Science 49(3): 610–625.
8. Johnson, E. J. and A. Tversky. 1983. “Affect, Generalization, and the Perception of Risk,” Journal of Personality and Social Psychology 45(1): 20–31.
9. Kasperson, R. E., O. Renn, P. Slovic, H. S. Brown, J. Emel, R. Goble, J. X. Kasperson, and S. Ratick. 1988. “The Social Amplification of Risk: A Conceptual Framework.” Risk Analysis 8(2): 177–187.
10. Keltner, D., K. D. Locke, and P. C. Aurain. 1993. “The Influence of Attributions on the Relevance of Negative Feelings to Personal Satisfaction.” Personality and Social Psychology Bulletin 19(1): 21–29.
11. Lerner, J. S., R. M. Gonzalez, D. A. Small, and B. Fischoff. 2003. “Effects of Fear and Anger on Perceived Risks of Terrorism: A National Field Experiment.” Psychological Science 14(2): 144–150.
12. Marcus, G. E., W. R. Neuman, and M. Mackuen. 2000. Affective Intelligence and Political Judgment. Chicago, IL: University of Chicago Press.
13. Marcus, G. E., J. L. Sullivan, E. Theiss-Morse, and D. Stevens. 2005. “The Emotional Foundation of Political Cognition: The Impact of Extrinsic Anxiety on the Formation of Political Tolerance Judgments.” Political Psychology 26(6): 949–963.
14. Pham, M. T. 1998. “Representativeness, Relevance, and the Use of Feelings in Decision Making.” Journal of Consumer Research 25(2): 144–159.
15. Rosenboim, U. B., S. Shahrabani, and T. Shavit. 2011. “Emotions, Risk Perceptions, and Precautionary Behavior under the Threat of Terror Attacks: A Field Study Among Israeli College Students.” Journal of Behavioral Decision Making.
16. Sadler, M. S., M. Lineberger, J. Correll, and B. Park. 2005. “Emotions, Attributions, and Policy Endorsements in Response to the September 11th Terrorist Attacks.” Basic and Applied Social Psychology 27(3): 249–258.
17. Schwarz, N. and G. L. Clore. 1996. “Feelings and Phenomenal Experiences.” In E. Tory Higgins and Arie W. Kruglanski (eds.), Social Psychology: Handbook of Basic Principles. New York: Guilford Press, 433–465.
18. Schwarz, N. and G. L. Clore. 1983. “Mood, Misattribution, and Judgments of Well-Being: Informative and Directive Functions of Affective States.” Journal of Personality and Social Psychology 45(3): 513–523.
19. Schwarz, N., W. Servay, and M. Kumpf. 1985. “Attribution of Arousal as a Mediator of the Effectiveness of Fear–arousing Communications.” Journal of Applied Social Psychology 15(2): 178–188.
20. Siemer, M. and R. Reisenzein. 1998. “Effects of Mood on Evaluative Judgments: Influence of Reduced Processing Capacity and Mood Salience.” Cognition & Emotion 12(6): 783–805.
21. Skitka, L. J., C. W. Bauman, N. P. Aramovich, and G. S. Morgan. 2006. “Confrontational and Preventative Policy Responses to Terrorism: Anger Wants a Fight and Fear Wants ‘Them’ to Go Away.” Basic and Applied Social Psychology 28(4): 375–384.
22. Slovic, P. 1987. “Perception of Risk.” Science 236(4799): 280–285.

How to Faster Test a Device for Different Combinations of Parameters

Francisco Zapata[1] and Luis Gutierrez[2]

[1]Department of Industrial and Systems Engineering and
[2]Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA
fazg74@gmail.com, lcgutierrez@miners.utep.edu

Abstract. A device has to function properly under all possible conditions: e.g., for all temperatures within a given range, for all possible humidity values within a given range, etc. Ideally, it would be nice to be able to test a device for all possible combinations of these parameters, but the number of such combinations is often so huge that such an exhaustive testing is not possible. Instead, it is reasonable to check the device for all possible values of each parameter, for each possible pairs of values of two parameters, and, in general, for all possible combinations of values of k parameters for some k. For n parameters, a straightforward testing design with this property contains $O(n^k) \cdot N^k$ experiments, where N is the number of tested values of each parameter. We show that, by using a more sophisticated testing design, we can decrease the number of experiments to a much smaller number $O(\log^{k-1}(n)) \cdot N^k$.

1 Formulation of the Problem

It is important to test a device for different combinations of parameters. Many devices have to function correctly under many different values of the corresponding parameters: e.g., for temperatures within the given range, for pressure within the given range, for humidity within the given range, etc.

It is not possible to test all possible combinations of parameters. Ideally, we should test the device for all possible combinations of the corresponding parameters. However, often, such a testing is not realistic. For example, if we have 20 possible parameters, and we consider 10 possible values of each of these parameters, then testing all possible combinations would require an unrealistic amount of 10^{20} tests. Even in the idealized situation when each test takes 1 second, then, with $3 \cdot 10^7$ seconds in a year, this testing would require $3 \cdot 10^{12}$ years – longer than the lifetime of the Universe.

Solution: test for all pairs, or all triples, etc. Since we cannot test for all possible combinations of all the parameters, we need to test at least for all possible values of each parameter separately. In other words, we need to test the device for all

possible values of outside temperature, then test this device for all possible values of humidity, etc.

In this testing, we may overlook possible joint effect of two or more different parameters. To take such an effect into account, it makes sense to arrange the tests in such a way that for every two parameters, we test all possible combinations of values. Similarly, we may want to test in such a way that for every three parameters, we test all possible combinations of values, etc.; see, e.g., [1–4].

How to arrange such a test: first simple idea. For each parameter x_i, we have a range $[\underline{x}_i, x_i]$ of possible values. Let us assume that for each parameter, we test for N different values $x_{i1} < x_{i2} < \ldots < x_{iN}$. In this case, we need N experiments to test the device's behavior for all N values of each parameter.

If we simply want to test for all possible values of each parameter, then a straightforward idea is to first test all possible values of the first parameter x_1, then test all possible values of the second parameter x_2, etc., until we have tested all the parameters. If we denote by n the number of parameters, then this scheme requires $n \cdot N$ experiments.

If we want to test all possible pairs of parameters, then, for each of $\binom{n}{2}$ pairs of parameters, we test all possible N^2 pairs of values. This requires $\binom{n}{2} \cdot N^2$ experiments.

Similarly, if we want to test all possible triples of parameters, then for each of $\binom{n}{3}$ triples of parameters, we test all possible N^3 triples of values. This requires $\binom{n}{3} \cdot N^3$ experiments. In general, if we fix an integer k, and we want to test all possible combinations of values of each k parameters, then for each of $\binom{n}{k}$ k-tuples of parameters, we test all possible N^k combinations of values. This requires $\binom{n}{k} \cdot N^k$ experiments.

We can test faster than that. It is easy to see that the simple straightforward approach uses too many combinations of parameters, we can often use much fewer experiments.

For example, if we want to test all possible values of each parameter, then in the above straightforward approach, we perform $n \cdot N$ experiments. In reality, it is sufficient to perform only N experiments. Namely, in each experiment $j = 1, \ldots, N$, we take each parameter x_i to be equal to x_{ij}:

- in the first experiment, we select the first value of each of n parameters, i.e., use parameters $(x_{11}, \ldots, x_{n1})$;
- in the second experiment, we select the second value of each of n parameters, i.e., use parameters $(x_{12}, \ldots, x_{n2})$;
- ...

– in the j-th experiment, we select the j-th value of each of n parameters, i.e., use parameters $(x_{1j}, \ldots, x_{nj})$;
– ...
– finally, in the last (N-th) experiment, we select the N-th value of each of n parameters, i.e., use parameters $(x_{1N}, \ldots, x_{nN})$.

When we have many parameters $n \gg 1$, we then have $n \cdot N \gg N$, so this idea drastically decreases the number of necessary experiments – and thus, the testing time.

Comment. In some cases, we have different number of values N_i for different parameters x_i. In this case, in the straightforward approach, we need $\prod_{i=1}^{n} N_i$ combinations, but instead, we can simply use $N = \max(N_i)$ combinations: namely, we set $x_{ij} = x_{iN_i}$ when $j > N_i$. Thus, when all the values N_i are of the same order, we still get a drastic decrease in the number of experiments.

What we do in this paper. In this paper, we show that similar faster testing is possible when we test all possible pairs of parameters, all possible triples, etc.

2 New Testing Design: Main Idea and Step-by-Step Description

Let us formulate the problem in precise terms. The above description leads to the following definition.

Definition. *Let $n > 0$, $N > 0$, and $k > 0$ be positive natural numbers. The number n will be called the* number of parameters*, and the number N will be called the* number of values.

– *By an* experiment*, we mean a tuple of n integers $j_1, \ldots, j_n$, where $1 \le j_i \le N$ for all i. We say that in this experiment, we use the j_i-th value of the i-th parameter. An experiment will also be denoted by $(x_{1i_1}, \ldots, x_{nj_n})$.*
– *By a* testing design*, we mean a finite set of experiments.*
– *We say that a testing design* tests each combination of k parameters *if for every k-tuple $1 \le i_1 < \ldots < i_k \le N$ and for all k-tuples of integers $v_1, \ldots, v_k$, with $1 \le v_\ell \le N$, this testing design contains an experiment in which, for all ℓ from 1 to k, we use the v_ℓ-th value of the i_ℓ-th parameter.*

Main objective. Our main objective is minimize the required number of experiments.

The straightforward ideas leads to a design that tests each combination of k parameters and that consists of $\binom{n}{k} \cdot N^k$ experiments. As a function of the number n of parameters, this number of experiments is $O(n^k) \cdot N^k$.

For $n = k$, we need to test all N^k possible combinations of parameters, so we cannot have fewer than N^k test anyway. However, as the above case of $k = 1$ shows, we can try to minimize the factor depending on n.

Main Result. *For each k, there exists a testing design that tests each combination of k parameters and that consists of $O(\log^{k-1}(n)) \cdot N^k$ experiments.*

Discussion.

- For $k = 1$, we get the known fact that we need $O(N)$ experiments.
- For testing all possible pairs ($k = 2$), we need $O(\log(n)) \cdot N^2$ experiments. This is much smaller than $O(n^2) \cdot N^2$ experiments needed in the straightforward approach.
- For testing all possible triples ($k = 3$), we need $O(\log^2(n)) \cdot N^3$ experiments. This is much smaller than $O(n^3) \cdot N^3$ experiments needed in the straightforward approach.

Description of the new testing design: case of $k = 2$. Let $B = \lceil \log_2(n) \rceil \sim \log(n)$ be the number of bits needed to describe all the natural numbers from 0 to $n-1$. Let us enumerate the bit from lowest to the highest. Let us denote the b-th bit in the binary expansion of an integer i by $\mathrm{bit}_b(i)$. For example, for the binary number $i = 1011_2 = 11_{10}$:

- the first (lowest) bit is 1: $\mathrm{bit}_1(i) = 1$;
- the second bit is 1: $\mathrm{bit}_2(i) = 1$;
- the third bit is 0: $\mathrm{bit}_3(i) = 0$;
- the fourth bit is 1: $\mathrm{bit}_4(i) = 1$, and
- all the other bits are 0s: $\mathrm{bit}_b(i) = 0$ for all $b > 4$.

Our new testing design consists of B groups of experiments. Each of these groups consists of N^2 experiments, so that total number of experiments is indeed $O(\log(n)) \cdot N^2$. In the b-th group of experiments, for each pair of integers (f, s), $1 \le f \le N$ and $1 \le s \le n$:

- we set $j_i = f$ if $\mathrm{bit}_b(i-1) = 0$, and
- we set $j_i = s$ if $\mathrm{bit}_b(i-1) = 1$.

If we have two different integers $i_1 < i_2$, then $i_1 - 1 \ne i_2 - 1$, so at least one bit b in the binary expansions of $i_1 - 1$ and $i_2 - 1$ is different. Thus, for this bit b, the corresponding group of experiments tests all possible pairs (f, s).

Example: $n = 4$. For $n = 4$, we need $B = 2$ bits to represent integers 0, 1, 2, and 3. Here, $0_{10} = 00_2$, $1_{10} = 01_2$, $2_{10} = 10_2$, and $3_{10} = 11_2$. Thus, in this case, we have two groups of N^2 experiments each:

- In the first group of experiments, we assign s to all the values i for which $\mathrm{bit}_1(i-1) = 0$, and f to all the values i for which $\mathrm{bit}_1(i-1) = 1$. Thus, each experiment has the form (f, s, f, s).
- In the second group of experiments, we assign s to all the values i for which $\mathrm{bit}_2(i-1) = 0$, and f to all the values i for which $\mathrm{bit}_2(i-1) = 1$. Thus, each experiment has the form (f, f, s, s).

If $i_1 < i_2$ are both odd or both even, then the second group of experiments tests all possible combinations of the values of the corresponding parameters. If one of the values i_1 and i_2 is odd and another value is even, then the first group of experiments tests all possible combinations of values.

Description of the new testing design: case of $k > 2$. To describe the testing design for $k > 2$, we use the following recursive algorithm that reduces a testing design for given k and n to a testing designs for smaller k and n.

For $n = k$, we just have to test all N^k possible combinations of values of all k parameters.

For $n > k$, we divide the set of n parameters into two halves of size $n/2$. Then:

- To cover situations when all k parameters are in the first half and situations when all k parameters are in the second half, we use the testing design for $n/2$ and k; each experiment in this design is copied for the second half, so, e.g., a design fs becomes $fsfs$ (see example below).
- To cover situations in which $k - 1$ parameters are in the first half and 1 parameter is in the second half, we combine each experiment from testing plan for $n/2$ and $k - 1$ with each experiment from the testing plan for $n/2$ and 1.
- To cover situations in which $k - 2$ parameters are in the first half and 2 parameters are in the second half, we combine each experiment from testing plan for $n/2$ and $k - 2$ with each experiment from the testing plan for $n/2$ and 2.
- ...
- To cover situations in which $k - i$ parameters are in the first half and i parameters are in the second half, we combine each experiment from testing plan for $n/2$ and $k - i$ with each experiment from the testing plan for $n/2$ and i.
- ...
- Finally, to cover situations in which 1 parameter are in the first half and $k - 1$ parameters are in the second half, we combine each experiment from testing plan for $n/2$ and $k - 1$ with each experiment from the testing plan for $n/2$ and 1.

References

1. R. Black, *Managing the Testing Process*, Wiley, New York, 2009.
2. D. R. Kuhn, Y. Lei, and R. Kacker, "Practical Combinatorial Testing: Beyond Pairwise", *IEEE IT Professional*, May/June 2008, Vol. 10, No. 3, pp. 19–23.
3. K. H. Pries and J. M. Quigley, *Testing Complex and Embedded Systems*, CRC Press, Boca Raton, Florida, 2010.
4. M. S. Reorda, Z. Peng, and M. Violante (eds.), *System-Level Test and Validation of Hardware/Software Systems*, Springer Verlag, London, 2005.

Selecting Strategies Based on Abstracted Game Models

Oscar Veliz and Christopher Kiekintveld
osveliz@miners.utep.edu,cdkiekinveld@utep.edu

University of Texas at El Paso
El Paso, TX, USA

Abstract. Solving large games is a key research challenge for real world applications of game theory. An important technique is using abstraction to simplify the game before performing further analysis, such as finding an equilibrium. For example, this approach has been instrumental in designing successful Poker-playing agents. We consider the problem of abstraction in general normal-form games, and in particular, how an agent should select a strategy based on an abstracted game model. While most previous work focuses on using abstraction to find equilibrium solutions, we show that playing according to the equilibrium of an abstracted game can lead to very poor outcomes. We present an extensive experimental analysis of abstraction in normal form games that includes several different forms of abstraction and several different solution concepts for analyzing the abstracted games. Our experiments show that robust solution methods outperform Nash equilibrium when reasoning based on abstracted games.

1 Introduction

Solving large games is one of the central research challenges in computational game theory [4]. Progress on scalable algorithms for game-theoretic analysis has proven important for real world applications such as homeland security [7]. Applying game theory to these problems requires formulating and analyzing very large game. Abstraction has emerged as an important technique for scaling game theory. For example, recent work on designing Poker agents competitive with humans has relied heavily on advances in automated abstraction methods for games [5]. The basic approach is to apply abstraction to drastically shrink the size of the game, and then to apply state of the art algorithms to find solutions to the abstracted games. A reverse mapping algorithm is use to map the solution back into a strategy that can be played in the original, unobstructed game.

We take the perspective of a player trying to use an abstracted game model to choose a strategy to play; this is known as the *strategy selection problem*. There is a fundamental conceptual problem with using a Nash equilibrium to select a strategy to play. The abstracted game is an imperfect representation of the original game, and it has been shown that Nash equilibrium is not robust to uncertainty about the game model [1]. We hypothesize that the success of Nash equilibrium combined with abstraction in games such as Poker is closely ties to the zero-sum properties of the game.

We investigate strategy selection problems with abstraction empirically using the framework of *meta-games* [1]. In the abstraction meta-game, each player receives a different abstraction of the original game and selects a strategy based on this abstraction.

The results of the strategy selections evaluated in the context of the original game. Using this framework, we run experiments with several different kinds of abstractions on both general-sum and zero-sum games, and evaluate several different methods for strategy selection. We show that Nash equilibrium can lead to unbounded losses when using abstraction, and that in general if performs poorly as a strategy selection method.

2 Abstracting Normal Form Games

A finite normal-form game is defined by a tuple (N, A, u), where N is a set of n players, $A = A_1 \times \cdots \times A_n$ is the set of actions, with A_i denoting the set of actions available to player i, and $u = (u_1, \ldots, u_n)$ represent the utility (payoff) functions for each player [6]. Each utility function $u_i : A \mapsto \mathbb{R}$ gives a real-valued payoff for player i for every possible combination of actions. Actions are also called pure strategies. Mixed strategies are defined as the set of possible probability distributions over a player's pure strategies, denoted by $S = S_1 \times \cdots \times S_n$. For a mixed strategy s_i we use $s_i(a_i)$ to denote the probability of playing action a_i. We define the expected utility (payoff) for player i for a profile of mixed strategies $s = (s_1, \ldots, s_n)$ as:

$$u_i(s) = \sum_{a \in A} u_i(a) \prod_{j=1}^{n} s_j(a_j) \qquad (1)$$

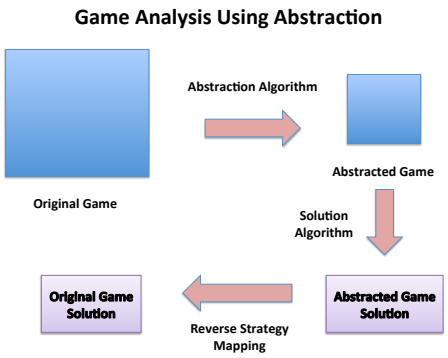

Fig. 1. Solving a game using abstraction.

We write $s = (s_i, s_{-i})$ to refer to the profile where player i plays s_i and the other players play according to the profile s_{-i}. A *best response* strategy for player i to a profile of opponent strategies s_{-i} is a strategy s_i^* such that $u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i}) \forall s_i \in S_i$. We define the *benefit to deviating* for player i from strategy profile s is defined as $\epsilon_i(s) = \max_{s_i' \in (S_i \setminus s_i)} (u_i(s_i', s_{-i}) - u_i(s_i, s_{-i}))$. The overall benefit to deviating from a strategy profile is the maximum benefit to deviating for any player, $\epsilon(s) = \max_{i \in N} \epsilon_i(s)$. A *Nash equilibrium* is a strategy profile s such that no player has a positive benefit to deviating, so $\epsilon(s) \leq 0$. If a strategy profile has a positive value of $\epsilon(s)$ it is as an approximate ϵ-Nash equilibrium.

An abstraction of a game simplifies the original game representation in some way. Figure 1 shows how the process works from the perspective of a single player. First, an abstraction algorithm is applied to compute the abstracted game representation. Then, a game analysis algorithm (such as an equilibrium solver) is applied to analyze the abstracted game and produce a solution. Since the strategy space of the abstracted game may not match the original game, a reverse mapping algorithm is applied to map back to strategies in the original game. There are many different approaches for abstracting

a game, and out purpose is not to find the best abstraction, but rather to investigate the implications of abstraction for strategic reasoning. Therefore, we start by studying two basic methods for abstracting games.

Random Removal: This abstraction method randomly removes a subset of the strategies for each player from the original game; the payoffs remain the same for the remaining strategy profiles. The reverse mapping uses the same probability for each action that is not removed during abstraction, and 0 for any removed strategies.

Payoff Bucketing: This abstraction method focuses on payoffs rather than the actions by reducing the granularity of the payoffs. For a given number of buckets b we divide the full range of possible payoffs $u_{range} = \max_{s,i} u_i(s) - \min_{s,i} u_i(s)$ into b equally-sized intervals, each of size $\frac{u_{range}}{b}$. Then, we map each payoff in the original game to the corresponding interval and set the payoff in the abstracted game to the midpoint of this interval. The set of actions is identical to the original game.

3 Nash Equilibrium in Abstracted Games

We now consider the problem of selecting strategies to play using abstracted game models. One approach for selecting a strategy is to solve the abstracted game for a Nash equilibrium (or approximation), and to play the equilibrium strategy. However, it is not obvious that choosing based on an equilibrium analysis will lead to desirable outcomes, since the equilibrium strategy is only a best response if the opponent's strategy is correctly predicted by the equilibrium.

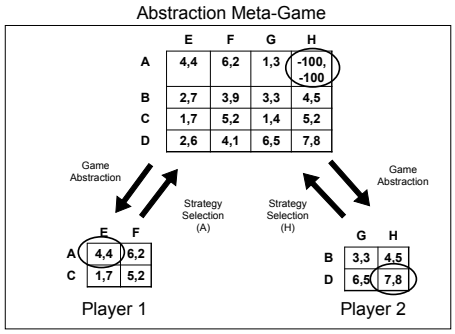

Fig. 2. The meta-game when two players use different abstractions.

In fact, we can construct examples where playing according to the Nash equilibrium of an abstracted game can lead to unbounded losses. Consider the meta-game [1] in Figure 2. There are two players in the original game, with four actions for each player. Both players apply random removal abstraction, reducing the game from 4×4 to 2×2 actions. It is important to note that the players do *not* necessarily compute the same abstracted game in our model. The players must reason based on different views of the game, and they may not know exactly what abstracted game the other players are using. The players reason about the abstracted games, select strategies to play, map the strategies back into the original game, and receive the corresponding payoffs in the original game.

For the example in Figure 2, both players play according to a Nash equilibrium of their abstracted game (the circled profiles in the small games). These are the *only* pure-strategy equilibria of the abstracted games, and they are based on dominant strategies. However, when the actions are played in the original game, the players choose strategy

profile (A, H), which is *not* an equilibrium of the original game, and is by far the worst outcome for both players. We could replace the payoffs of -100 with arbitrarily large negative values, so the potential loss to the players is unbounded.

4 Game Analysis Methods

In the previous section we demonstrated that playing according to a Nash equilibrium may not result in good outcomes for players reasoning based on abstracted games. We now introduce several other methods for selecting strategies which take as input an abstracted game and a player, and return a mixed strategy for the player.

Uniform Random (UR): This method always returns the uniform random mixed strategy that selects all actions with equal probability.

Best Response to Uniform (BRU): This method assumes that all players other then the selected player will play the uniform random strategy, and plays a best response to this uniform distribution.

Logistic Quantal Response Equilibrium (LQRE): Quantal Response Equilibrium (QRE) [3] modifies the standard notion of Nash equilibrium by replacing the best response function with a noisy best response function. In a QRE, the strategies are noisy best responses to one another. The most common form of QRE uses a logistic distribution to specify the noisy best response in the following form:

$$s_i(a_i) = \frac{\exp(\lambda \cdot u_i(a_i, s_{-i}))}{\exp(\sum_{a'_i \in A_i} \lambda \cdot u_i(a'_i, s_{-i}))} \tag{2}$$

The parameter λ specifies the amount of noise in the distribution. As $\lambda \mapsto 0$ the response approaches a uniform random strategy, and as $\lambda \mapsto \infty$ the response approaches the best response function (and therefore LQRE converges to Nash). We compute LQRE using Gambit [2], specifically the *gambit-logit* solver [8] . The strategy selected by our method is a best response to the LQRE, so we assume noise only for predicting the other player's choices.

Nash Equilibrium (NE): This method plays according to a mixed strategy equilibrium of the abstracted game. We again use Gambit [2] to solve for a mixed strategy Nash equilibrium of the game.

Approximate Pure Strategy Nash Equilibrium (APSNE): This method focuses on the pure strategy profiles of the game, and selects a strategy according to the most stable profile (the one with the smallest benefit to deviating, $\epsilon(s)$). In cases where there is at least one pure strategy Nash equilibrium, this will select one of these Nash equilibria that has the largest loss for deviating.

References

1. C. Kiekintveld and M. Wellman. Selecting strategies using empirical game models: An experimental analysis of meta-strategies. In *Seventh International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1095–1102, 2008.
2. R. D. McKelvey, A. M. McLennan, T. L. Turocy, and http://www.gambit project.org. Gambit: Software tools for game theory. Technical Report Version 13.1.0, 2013.

3. R. D. McKelvey and T. R. Palfrey. Quantal response equilibrium for normal form games. *Games and Economic Behavior*, 10(1):6–38, 1995.
4. N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007.
5. T. Sandholm. The state of solving large incomplete-information games, and application to poker. *AI Magazine*, Special Issue on Algorithmic Game Theory, 2010.
6. Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, Feb. 2009.
7. M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.
8. T. L. Turocy. A dynamic homotopy interpretation of the logistic quantal response equilibrium correspondence. *Games and Economic Behavior*, 51:243–263, 2005.

COPROD v2.0 13

Constraint programming techniques are important components of intelligent systems. They constitute a nice and easy way to represent many practical situations. They have been successfully applied to a number of fields, such as scheduling of air traffic, software engineering, networks security, chemistry. Despite their proven usefulness, they are still under-utilized in real-life projects. One reason might be the lack of effective communication of researchers with domain practitioners about constraints in general and their use in decision making in particular.

CoProD aims at easing collaborations and therefore the emergence of new techniques. It aims to draw together a network of researchers interested in constraint and decision-making techniques, in particular researchers and practitioners that use numeric and symbolic approaches (or a combination) to solve constraints and optimization problems; and to address the gap between the great capacity of these techniques and their limited use.

This booklet compiles the abstracts of the CoProD'13 speakers: researchers in constraint programming, decision making, but also domains scientists, on topics such as optimization, interval computations, uncertainty.

CoProD'13 is the sixth edition of CoProD.