2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)

Towards the Detection of Partial Feature Interactions

Byron DeVries
School of Computing and Information Systems
Grand Valley State University
Allendale, Michigan
Email: devrieby@gvsu.edu

Betty H.C. Cheng
Department of Computer Science and Engineering
Michigan State University
East Lansing, Michigan
Email: chengb@cse.msu.edu

***Abstract*—Discrete feature interactions occur when the presence of overlapping, yet conflicting, features impact the functionality of an entire system such that one or more features are unsatisfied. However, adaptive systems may mitigate uncertainty by composing features from requirements that may be partially satisfied or partially unsatisfied (i.e., satisficed). In contrast to discrete interactions, these satisficed features that are composed of satisficed requirements may be partially unsatisfied due to a feature interaction. Feature interactions, in the case of satisficement, result in a proportional trade-off between conflicting objectives rather than a complete, or discrete, failure of features or the system. In this paper, we propose a new concept, *partial feature interactions*, to represent the class of feature interactions that allow for satisficement trade-offs between conflicting features. We also describe a method to reduce the dimensionality of the satisficement trade-offs to be reviewed when assessing the severity of the n-way interactions. Finally, we present a method for identifying Pareto optimal trade-offs among the conflicting features using the reduced dimensionality in order to detect *partial feature interactions* and multiple candidate adaptation points. We apply our method to an exemplar feature interaction from the automotive domain.**

I. Introduction

Feature interactions have been one of the fundamentally difficult issues in the development of systems, especially cyber-physical systems [1], [2]. Existing methods of detection, avoidance, and mitigation work to remove a feature interaction from the specification at design time or from the execution of an implemented specification at run time [1], [3], [4]. However, feature interactions are not intrinsically an "all-or-nothing" proposition where the discrete satisfaction of one set of features necessitates a "winner-take-all" approach. Instead, feature interactions can often be analyzed as conflicting objectives where a proportional satisfaction (i.e., satisficement) [5], [6] of one objective may allow for the partial satisficement of another objective. The trade-off between the satisficement of one or more features provides adaptation points for adaptive cyber-physical systems.

This paper introduces the concept of *partial feature interactions* and a supporting methodology. Specifically, we describe a set of methods to i) reduce the dimensionality of the trade-offs to be reviewed for n-way *partial feature interactions* and ii) identify Pareto optimal trade-offs between conflicting objectives within the features involved in an interaction in order to identify the Pareto optimal configurations of the system in the detected *partial feature interaction*. The set of Pareto optimal configurations enables the system to adapt if a prioritized feature is unacceptably impacted by the *partial feature interaction* by providing a suite of possible adaptation solutions known to improve the satisficement of the prioritized feature.

Design-time detection of feature interactions typically requires an exponential effort and possible exponential number of results to be reviewed involving all possible feature combinations [3], [7]. Rather than calculate and return the trade-off for every possible feature combination, we propose to analyze each feature for its ability to *cause* a feature interaction. That is, we analyze each feature for its satisficement trade-off with any subset of the remaining features. More formally, given a set of features, F, we analyze the trade-off of each feature f (i.e., $\forall f \in F$) against any set, S, of remaining features (i.e., $\forall S \subseteq (F \backslash \{f\})$), where an interaction exists within a set of features, $S \cup \{f\}$, and the feature, f, if a trade-off exists. Therefore, we reduce the number of trade-offs returned to be linear with respect to the number of features (i.e., $|F|$)[1] *and* limit the dimension of each trade-off analyzed to two objectives.

The contributions of this paper are as follows:

- We introduce the term *partial feature interactions*,
- We present a method to reduce the dimensionality of the trade-offs between conflicting requirements, and
- We present a method to identify the Pareto optimal trade-offs between conflicting features using not only a reduced number of trade-offs [7], [8], but also a *reduced dimensionality* of the trade-offs and demonstrate it on an interaction within the automotive domain.

We illustrate our approach with an example from the automotive domain. The remainder of this paper is organized as follows. Background information is provided in Section II. The approach of our method and an illustrative example are covered in Sections III and IV, respectively. Work related to the methods presented in this paper are compared in Section V. Finally, our conclusions and plans for future work are described in Section VI.

II. Background

This section overviews feature interactions, multi-objective optimization, and the satisficement of requirements.

[1]Note: The *computational* cost of calculating each trade-off is still exponential, even though the number of trade-offs to be analyzed is reduced.

978-1-7281-3368-3/19/$31.00 ©2019 IEEE
DOI 10.1109/SEAMS.2019.00026

A. Feature Interactions

Feature interactions have long been studied, especially in the context of telecommunication systems [1]. However, they have been studied assuming that features fail or succeed discretely, such that a feature interaction can be defined as the failure to achieve the composed properties of a set of composed features. For example, Calder, *et al.* [4] described a feature interaction between two features (F_1 and F_2) and their respective properties (ϕ_1 and ϕ_2). The description is as follows. A feature, F_1, satisfies a property, ϕ_1, denoted as $F_1 \models \phi_1$. Features may be combined, or composed, via a composition operator ('$\oplus$'). Therefore, features F_1 and F_2, may be composed as $F_1 \oplus F_2$. When $F_1 \models \phi_1$ and $F_2 \models \phi_2$, the expected composition of F_1 and F_2 should satisfy the conjunction of their respective properties, that is, $F_1 \oplus F_2 \models \phi_1 \wedge \phi_2$. However, if the composition of the features does not satisfy the conjunction of their respective properties (i.e., $F_1 \oplus F_2 \not\models \phi_1 \wedge \phi_2$), then there exists a feature interaction [4]. This 2-way interaction concept can be extended to represent n-way interactions where any number of composed features result in the satisfaction of their aggregate properties [7]:

$$\bigoplus_{i=1}^{|F|}(F_i) \not\models \bigwedge_{i=1}^{|F|}(\phi_i), \qquad (1)$$

where F is a set of features.

B. Multi-Objective Optimization

Multi-objective optimization [9] can be described as a comparison to single-objective optimization. In the case of single-objective optimization, there is one objective that is to be minimized or maximized to provide optimal utility. For example, if a customer is interested in buying a car with the fastest top speed then any possible car could be compared, and the car with the fastest top speed would be selected. Multi-objective optimization involves not just one objective (e.g., the top speed of a car) but multiple objectives. For example, if a customer is interested in buying a car with the fastest top speed *and* the longest range, it is unlikely that the fastest car would also have the longest range due to, among other factors, the weight of fuel. Similarly, it is also unlikely that the car with the longest range would also have the highest top speed. Therefore, it is impossible to fully optimize both objectives due to their conflicting conditions.

In order to identify the optimal choices when more than one objective exists, the trade-offs between the objectives must be taken into account. Given a preference for the range, a slow car with a long range may be selected. Conversely, given a preference for speed, a fast car with short range may be selected. Both are optimal *given a known preference* between the two objectives. Rather than a single optimum, conflicting objectives in multi-objective optimization may result in a set of optima. That is, there is a set of optima that can only improve a single objective if another objective is diminished. This set of optima is the Pareto optimal set or the *Pareto front* [9] representing the optimal set of trade-offs between the conflicting objectives in multi-objective optimization.

C. Satisficement

Satisfaction of requirements, while typically specified as a Boolean expression, may also be represented as a degree of satisfaction called *satisficement* [5], [6], [10]. One method of introducing this degree of satisfaction, or satisficement, into textual requirements is with the RELAX language [10] that introduces flexibility via temporal (e.g., 'close to' and 'as soon as possible') or ordinal (e.g., 'as many' and 'as few') RELAXation [5]. While the RELAX language is used in this paper, any method of describing satisficement of requirements may be substituted (e.g., FLAGS [11], or softgoals in *i** [12] and Tropos [13]).

Regardless of the method used to describe the requirements, the satisfaction is not measured discretely, but, instead, is measured as satisficement in an amount proportional to the degree the intent of the requirement is satisfied. For example, consider the following RELAXed requirement: "The adaptive cruise control **shall** maintain the current speed *as close as possible* to desired speed." An adaptive cruise control system may not always maintain the desired speed when an obstacle prevents the desired speed from being achieved without collision. However, while meeting the desired speed would completely satisfy the requirement, it is intuitive to consider a reduction in speed to at least partially satisfy the requirement.

III. Approach

An overview of the partial feature interaction analysis process defined in this paper is as follows. First the number of trade-offs and the dimensionality of each trade-off is reduced. Second, the reduced set of trade-offs is analyzed for conflicting objectives (i.e., feature interactions). Finally, the Pareto-optimal sets of the conflicting features are stored as candidate adaptation points.

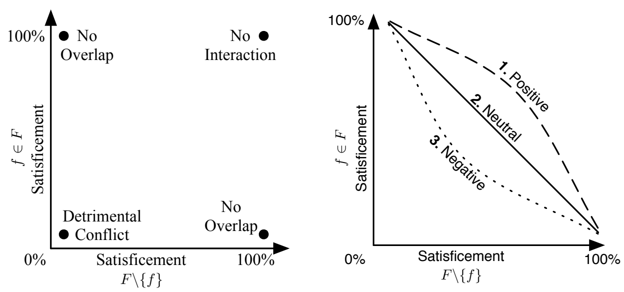
Reduce Trade-Offs & Dimensionality:

Detecting interactions typically induces an exponential effort with an exponential number of possible feature combinations [3]. However, analyzing the trade-off of any feature with any set of the remaining features reduces possible feature combinations to the number of features (i.e., linear with respect to the number of features), though feature interaction detection analysis effort remains exponential [7]. For example, given a set of four features (e.g., F_1, F_2, F_3, and F_4), rather than analyze the trade-offs among all 11 possible combinations of features (e.g., $\{F_1, F_2\}$, $\{F_1, F_3\}$, $\{F_2, F_3\}$, $\{F_1, F_2, F_3\}$, $\{F_1, F_4\}$, $\{F_2, F_4\}$, $\{F_1, F_2, F_4\}$, $\{F_3, F_4\}$, $\{F_1, F_3, F_4\}$, $\{F_2, F_3, F_4\}$, and $\{F_1, F_2, F_3, F_4\}$), each feature is analyzed for trade-offs against the worst case of the remaining features (e.g., F_1 vs a subset of $\{F_2, F_3, F_4\}$, F_2 vs a subset of $\{F_1, F_3, F_4\}$, F_3 vs a subset of $\{F_1, F_2, F_4\}$, and F_4 vs a subset of $\{F_1, F_2, F_3\}$). Given the four features, this strategy reduces the number of analysis results from eleven to four *and* reduces the worst case trade-off dimensionality with respect to the number of features, from linear to two, in all cases. The reduction in the number of trade-offs and the dimensionality of the trade-offs results in a linear number of trade-offs with a dimensionality of each trade-off limited to two objectives. Note that the trade-off is identified by feature interaction detection analysis and the Pareto optimal points are calculated from the possible trade-offs where the dimension of the trade-offs significantly increases the dimensionality of the Pareto front.

Analyze Trade-Offs:

Since feature interaction satisficement trade-offs can be assumed to be between a feature (i.e., $f \in F$) and some combination of the remaining features (i.e., a subset of $F \backslash \{f\}$), all Pareto optimal fronts can be defined with only these two objectives (i.e., a feature and a subset of the remaining features). Discrete features, or those that either fail or succeed, fall into one of the three categories labeled in Figure 1(a). First, there may be *No Overlap* between conflicting objectives. That is, one feature or combination of features is satisfied, causing the other to be unsatisfied. In this case, the Pareto optimal set only includes these two points. Second, there may be *Detrimental Conflict* amongst the objectives meaning the features conflict so extensively that neither can ever be satisfied. Finally, there may be *No Interaction* between the objectives. All features can be satisfied creating a Pareto optimal set of a single fully optimal point.

These cases, however, all represent traditional discrete states of satisfaction and do not take into account how satisficement may allow for partial feature interactions. Figure 1(b) identifies three Pareto fronts where a continuous trade-off between the satisficement of a feature (i.e., $f \in F$) and some combination of the remaining features (i.e., a subset of $F \backslash \{f\}$) exist due to a partial feature interaction.



(a) Discrete Trade-Offs

(b) Continuous Trade-Offs

Fig. 1: Feature Tradeoffs

These three categories are as follows:

1. *Positive Trade-Offs* indicate the Pareto front is convex due to a beneficial trade-off for both objectives (i.e., the feature under analysis and the combination of the remaining features). For example, if a point exists where increasing the top speed of the car requires giving up a greater proportion of fuel economy *and* increasing fuel economy requires giving up a greater proportion of the top speed, then we have a positive trade-off of satisficement.

2. *Neutral Trade-Offs* are when the Pareto front is linear indicating a linear trade-off regardless of the increase or point on the Pareto front. For example, if increasing the top speed of the car requires giving up the same proportion of fuel economy regardless of the point on the Pareto front, then we have a neutral trade-off of satisficement.

3. *Negative Trade-Offs* refer to the Pareto front that is concave due to a disadvantageous trade-off for both objectives. For example, if a point exists where increasing the top speed of the car requires giving up a lesser proportion of fuel economy *and* increasing fuel economy requires giving up a lesser proportion of the top speed, then we have a negative trade-off of satisficement and will likely move towards an extreme.

In each of these three trade-off scenarios, an infinite number of Pareto optimal solutions exist along the Pareto front. However, due to the partial feature interaction, no solutions can be identified above and to the right of the Pareto optimal sets. That is, the overall satisfaction of the system is negatively impacted by the interaction [14], and any trade-off scenarios only allow different feature satisficements in an ultimately unsatisfied, or satisficed, system. Note, satisficement differs significantly from discrete satisfaction where no solutions that partially satisfy (i.e., satisfice) both features can be identified.

Store Adaptation Points:

While the Pareto optimal set can be used as a traditional counterexample indicating a feature interaction that should be resolved by the system designer, the Pareto optimal set can also be stored for later use. Adaptive systems, for example, could use the Pareto optimal set to identify adaptations to configurations that are more highly satisficed for a prioritized feature. A simple example is described next.

IV. Example

This section provides the results (e.g., trade-offs) of applying the partial feature interaction detection process to an example set of RELAXed features.

A. Example System

An Adaptive Cruise Control (ACC) system for an automobile is used as an example. We define four features related to the ACC system:

F_1**:** Maintain Desired Speed
F_2**:** Minimize Fuel Consumption
F_3**:** Stay Below Speed Limit
F_4**:** Minimize Travel Time

Each of these features is defined by a decomposition of requirements. For example, the decomposition of F_1 includes requirements for setting the desired speed, measuring the actual speed, and adjusting the throttle to minimize the difference between the desired and actual speed. While the definition of features has numerous overlapping properties [15], each of these features (i.e., F_1, F_2, F_3, and F_4) is *at least* "an increment in product functionality" [16] for an ACC system.

The features themselves are represented by high-level RELAXed goals that represent the satisficement of composed requirements in order to address uncertainty in adaptive systems [5], [6], [17]. In this case, we assume the most fuel efficient speed is 40 miles per hour (MPH), the speed limit is 60 MPH where both higher and lower speeds are less desired, the desired speed set by the driver is 75 MPH due to an impatient driver, who is running late for an appointment, and the speed for the shortest trip is the highest available (100 MPH). Figure 2 illustrates the satisficement of the features based on the speed of the vehicle itself. Importantly, the features are fully satisficed (i.e., feature satisficement is value '1.0') at different speeds. For example, F_1 is fully satisficed when the speed of the vehicle is at 75 MPH (i.e., the desired speed), while F_2 is fully satisficed when the speed of the vehicle is the most fuel efficient speed (i.e., 40 MPH). The triangular graphs represent the degree of satisficement given a speed in the range of 0 MPH to 100 MPH where the peak satisficement for each feature is its target speed.

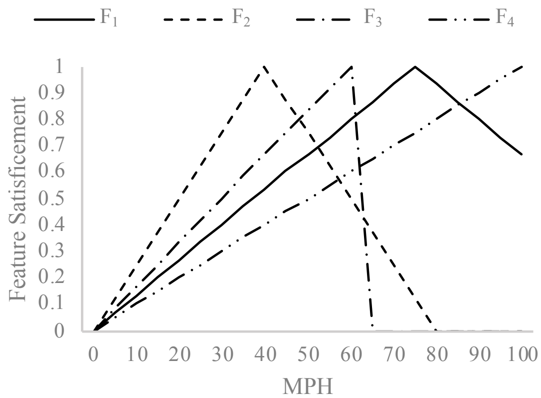


Fig. 2: Feature Satisficement

Given the difference in optimal speeds for each feature, it is clear that all features cannot be fully satisficed simultaneously. That is, improving the satisficement of one feature will necessitate reducing the satisficement of another feature.

B. Pareto Optimal Trade-Offs

In order to find the optimal trade-off for each feature, the Pareto optimal front must be identified for each feature when compared to the remaining features. In the case when there are four features, the comparison is simply one feature's trade-off with the worst case combined satisficement of the other three. The trade-offs, or Pareto optimal fronts, are as follows:

- F_1 with the worst case subset of F_2, F_3, and F_4
- F_2 with the worst case subset of F_1, F_3, and F_4
- F_3 with the worst case subset of F_1, F_2, and F_4
- F_4 with the worst case subset of F_1, F_2, and F_3

The satisficement for each feature (i.e., F_1, F_2, F_3, and F_4) is shown in Figure 2. The trade-off for each feature is analyzed against the worst-case subset of the remaining features. The worst-case subset for our example is any composed feature set that includes the feature in the remaining set of features with the lowest satisficement for the given environmental scenario (i.e., speed).

Given the individual feature satisficement and the fact that the composed worst case subset of the remaining features will have the satisfaction of the minimally satisfied feature for a given environmental scenario, we can graph the Pareto fronts for the trade-offs of each feature in Figures 3, 4, 5, and 6. These trade-off graphs plot the satisficement for the feature under analysis (y-axis) and the worst case subset of the remaining features for speeds (x-axis) for satisficement values when analyzed at 0 MPH to 100 MPH, in increments of 5 MPH. In Figures 3, 4, 5, and 6, the Pareto front, or points where no other points has better satisficement for both the feature under analysis and the worst case subset of the remaining features, is shown in black. Points that are dominated by the Pareto front, or are known to be worse than another existing point, are shown in gray. The speed in miles per hour (i.e., environmental scenario) is labeled for each point along the Pareto front, and the line connecting the points indicates connections to the higher and lower speeds in 5 MPH increments and decrements.

Analyzing feature F_1, *Maintain Desired Speed*, for a trade-off compared to the worst-case subset of the remaining features (i.e., F_2, F_3, and F_4) results in the Pareto front identified in Figure 3. Starting at the left, the first Pareto-optimal point occurs at 75 MPH where the feature under analysis (i.e., F_1) is fully satisfied on the x-axis. However, the worst case subset of the remaining features on the x-axis has no satisficement due to feature F_3 (i.e., *Stay Below Speed Limit*) since the 60 MPH speed limit is violated. As the speed decreases, the worst case subset of the remaining features on the y-axis increases satisficement as the speed limit is no longer violated. However, at 50 MPH, decreasing the speed no longer increases the satisficement of either feature F_1 or the worst case subset of the remaining features, as indicated by the gray color.

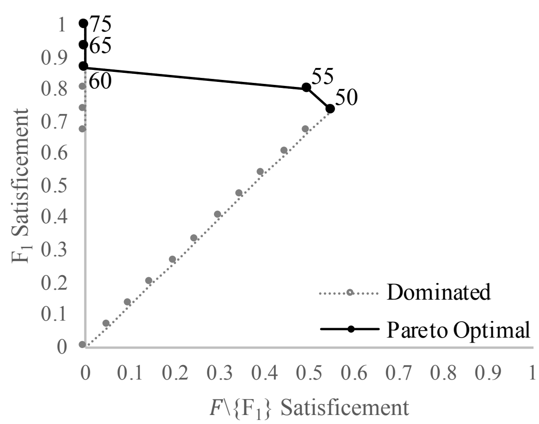


Fig. 3: Feature F_1 Trade-Off Graph

Analyzing feature F_2, *Minimize Fuel Consumption*, on the y-axis for a trade-off compared to the worst-case subset of the remaining features (i.e., F_1, F_3, and F_4) on the x-axis results in the Pareto front identified in Figure 4. The first Pareto-optimal point occurs at 40 MPH, the most fuel efficient speed, on the left and continues through to 60 MPH on the right.

Analyzing feature F_3, *Stay Below Speed Limit*, on the y-axis for a trade-off compared to the worst-case subset of

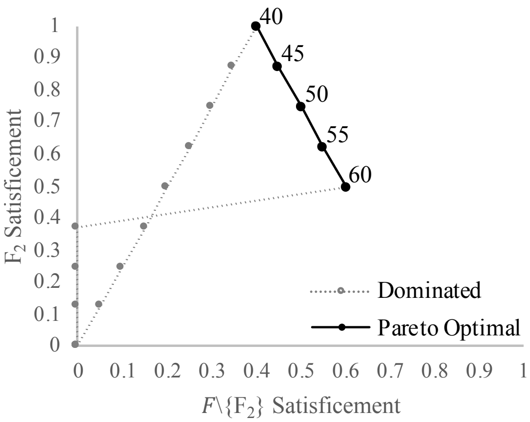


Fig. 4: Feature F_2 Trade-Off Graph

the remaining features (i.e., F_1, F_2, and F_4) on the x-axis results in the Pareto front identified in Figure 5. The only two Pareto-optimal points identified are at 60 MPH and 55 MPH. Speeds greater than 60 MPH fail to provide any satisficement to feature F_3, as the speed limit (i.e., 60 MPH) is violated.

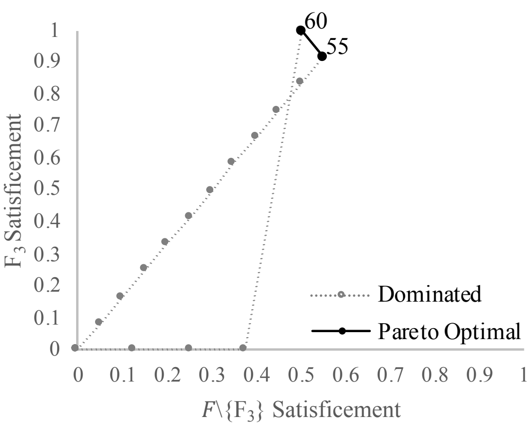


Fig. 5: Feature F_3 Trade-Off Graph

Analyzing feature F_4, *Minimize Travel Time*, on the y-axis for a trade-off compared to the worst-case subset of the remaining features (i.e., F_1, F_2, and F_3) on the x-axis results in the Pareto front identified in Figure 6. The feature under analysis, feature F_4, is most highly satisficed at the maximum speed analyzed (i.e., 100 MPH). However, just like the analysis of feature F_1, when the speed is greater than the speed limit, the worst-case of the remaining features is entirely unsatisfied due to the speed limit violation in feature F_3. Once the speed limit is no longer violated (i.e., at 60 MPH), the worst case of the remaining features improves while the speed reduces to 55 MPH. Speeds lower than 55 MPH means that neither the feature under analysis nor the worst-case subset of the remaining features improve (as indicated by the grayed-out dots and line).

C. Discussion

The partial feature interactions in Figures 3, 4, 5, and 6 are due to conflicting objectives represented by the satisficement

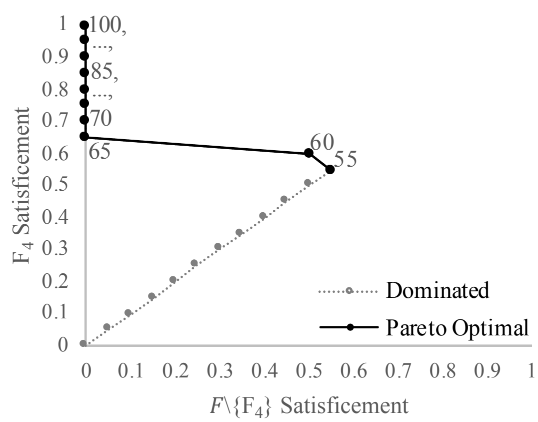


Fig. 6: Feature F_4 Trade-Off Graph

functions for each feature in Figure 2 and the Pareto optimal points are stored for future use as candidate adaptation points. In all cases, maximizing for a single feature lowers the satisficement of the remaining features. The optimal environmental scenario for each feature is identified when the feature is under analysis individually (i.e., 75 MPH for feature F_1, 40 MPH for feature F_2, 60 MPH for feature F_3, and 100 MPH for feature F_4). For example, F_1 in Figure 3 is fully satisficed when the desired speed is 75 MPH. However, the worst-case subset of the remaining features are completely unsatisficed (i.e., satisficement is value '0.0') due to F_3's lack of satisficement when the speed limit (60 MPH) is exceeded. The area above and to the left of each of the Pareto front is unreachable due to the partial feature interactions. However, given a strong preference between conflicting features, it may be desirable to maximize the preferred feature regardless of the impact on the remaining features. When a preference between conflicting features does not exist, maximizing the solution to achieve the highest overall system satisficement (i.e., maximizing the least satisficed feature) may be desirable. For example, in the case of the Pareto front for the analysis of F_1 in Figure 3, the rightmost point $(0.55, 0.7\overline{3})$ gives the highest composed satisficement of the system (i.e., minimum of the conflicting composed features) at 0.55 via a speed of 50 MPH.

D. Possible Use of Stored Adaptation Points

Stored adaptation points can be used in adaptive and autonomic systems via the Monitor-Analyze-Plan-Execute over a shared Knowledge (MAPE-K) feedback loop [18]. During execution, adaptive cyber-physical systems may be required to adapt due to unforeseen circumstances. For example, an electric vehicle with an ACC system may initially prioritize staying below the speed limit (i.e., feature F_3). Based on the expected driving distance to the destination and the originally designed battery capacity, MAPE-K analysis and planning phases early in the trip may indicate that such a prioritization will enable the electric vehicle to travel to its destination within its current electrical charge *and* up to the speed limit along the route.

In this example scenario, feature F_3 is prioritized. The analysis of feature F_3 in Figure 5 includes two Pareto optimal points, one at 55 MPH and one at 60 MPH. Feature F_3 (*Stay Below Speed Limit*) has a higher satisficement at 60 MPH, therefore 60 MPH is the selected speed without run-time trade-off analysis in the analysis phase of the MAPE-K loop. Instead, the optimal trade-offs are already known in the knowledge base of the MAPE-K loop.

A sensor may detect an unforeseen change in the amount of electricity available from the battery. Perhaps the batteries no longer hold the originally designed capacity *and* the temperature on the trip was significantly colder than usual, thus further reducing the output of the batteries. Analysis within the MAPE-K loop may determine that the current battery load will not support the travel time necessary to arrive at the planned destination and an adaptation is required.

Assuming the MAPE-K planned adaptation prioritizes minimizing fuel consumption (i.e., feature F_2), the optimal trade-offs have already been identified. Given a speed of 60 MPH, the Pareto front identified and stored for feature F_2 (as shown in Figure 4) indicates that while 60 MPH is a Pareto optimal point, feature F_2 is not fully optimized within the Pareto set. That is, we are not fully optimizing for fuel savings. As part of adaptation, the electric vehicle must adjust its speed from 60 MPH to 40 MPH (as shown in Figure 7) in order to maximize the available range by minimizing fuel consumption within the set of candidate Pareto optimal adaptations identified prior to execution. The red arrow indicates the speed transition from 60 MPH to 40 MPH in the MAPE-K planning phase based on stored Pareto optimal candidate adaptation points. Fuel consumption is minimized while simultaneously, ensuring a minimal trade-off with other requirements.

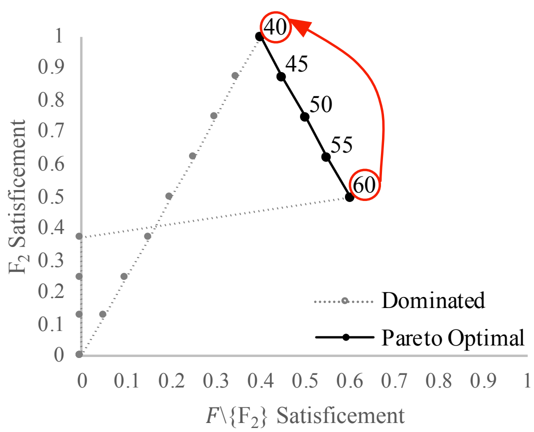


Fig. 7: Adaptation to Prioritize F_2

V. Related Work

Existing methods for detecting, avoiding, and resolving discrete feature interactions have been summarized in earlier surveys [1], [3], [4]. The concept of a trade-off, or impact from another feature or requirement, has also been considered with *i** where the system designer can specify non-functional trade-offs using contribution links comprising discrete measures of contribution [12]. Other fuzzy-logic based approaches also exist specifically for conflicting imprecise requirements, but do not reduce the dimensionality of the trade-offs or operate on features [19], [20].

The method presented in this paper differs from previous approaches in that it does not require trade-offs to be explicitly defined by the system designer, it operates on features rather than functional or non-functional requirements, and it provides the Pareto optimal trade-offs using continuous satisficement rather than a discrete satisfaction or discretely enumerated contribution links over a reduced (linear) number of trade-offs with reduced (two) dimensionality. Finally, the method presented in this paper stores the identified Pareto optimal trade-offs for future use in the adaptation process.

VI. Conclusions

In this paper, we have introduced *partial feature interactions* and methods for identifying Pareto optimal trade-offs between conflicting features after significantly reducing the dimensionality. This reduction decreases both the number of trade-offs that must be analyzed to be linear with the number of features, and also reduces the multi-objective optimization to only two objectives (i.e., the feature and any subset of the remaining features) instead of a more difficult many-objective problem. In the example presented, we reduce the trade-offs analyzed from eleven to only four, and reduce the dimension of each trade-off analyzed from four to only two. *Partial feature interactions* are an important advancement to the detection of feature interactions for systems that can be described using features with satisficement (e.g., adaptive cyber-physical systems) rather than discrete satisfaction. We have illustrated these methods on an automotive example using ACC that must adapt due to unforeseen circumstances.

Future work will focus on additional decision making criterion to choose appropriate trade-offs from the stored Pareto optimal set at run time and design time. These results can be used to further support the adaptation of cyber-physical systems in the face of uncertainty using early state requirements artifacts (e.g., requirements, goal models, and product lines).

Acknowledgments

This work has been supported in part by grants from NSF (CNS-1305358 and DBI-0939454), Ford Motor Company, and General Motors Research; and the research has also been sponsored by Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0284. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory (AFRL), the U.S. Government, National Science Foundation (NSF), Ford, General Motors (GM), Michigan State University (MSU), Grand Valley State University (GVSU), or other research sponsors.

REFERENCES

[1] S. Apel, J. M. Atlee, L. Baresi, and P. Zave, "Feature Interactions: The Next Generation (Dagstuhl Seminar 14281)," *Dagstuhl Reports*, vol. 4, no. 7, pp. 1–24, 2014. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2014/4783

[2] S. Apel, "The new feature interaction challenge," in *Proceedings of the Eleventh International Workshop on Variability Modelling of Software-intensive Systems*. ACM, 2017, pp. 1–1.

[3] S. Apel, S. Kolesnikov, N. Siegmund, C. Kästner, and B. Garvin, "Exploring feature interactions in the wild: the new feature-interaction challenge," in *Proceedings of the 5th International Workshop on Feature-Oriented Software Development*. ACM, 2013, pp. 1–8.

[4] M. Calder, M. Kolberg, E. H. Magill, and S. Reiff-Marganiec, "Feature interaction: a critical review and considered forecast," *Computer Networks*, vol. 41, no. 1, pp. 115–141, 2003.

[5] J. Whittle, P. Sawyer, N. Bencomo, B. H. C. Cheng, and J.-M. Bruel, "Relax: Incorporating uncertainty into the specification of self-adaptive systems," in *Requirements Engineering Conference, 2009. RE'09. 17th IEEE International*. IEEE, 2009, pp. 79–88.

[6] ——, "Relax: a language to address uncertainty in self-adaptive systems requirement," *Requirements Engineering*, vol. 15, no. 2, pp. 177–196, 2010.

[7] B. DeVries and B. H. C. Cheng, "Automatic detection of feature interactions using symbolic analysis and evolutionary computation," in *2018 IEEE International Conference on Software Quality, Reliability and Security (QRS)*. IEEE, 2018, pp. 257–268.

[8] ——, "Run-time monitoring of self-adaptive systems to detect n-way feature interactions and their causes," in *2018 IEEE/ACM 13th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. IEEE, 2018, pp. 94–100.

[9] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.

[10] J. Whittle, P. Sawyer, N. Bencomo, and B. H. C. Cheng, "A language for self-adaptive system requirements," in *Service-Oriented Computing: Consequences for Engineering Requirements, 2008. SOCCER'08. International Workshop on*. IEEE, 2008, pp. 24–29.

[11] L. Baresi, L. Pasquale, and P. Spoletini, "Fuzzy goals for requirements-driven adaptation," in *Requirements Engineering Conference (RE), 2010 18th IEEE International*. IEEE, 2010, pp. 125–134.

[12] E. S. Yu, "Towards modelling and reasoning support for early-phase requirements engineering," in *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*. IEEE, 1997, pp. 226–235.

[13] J. Castro, M. Kolp, and J. Mylopoulos, "Towards requirements-driven information systems engineering: the tropos project," *Information systems*, vol. 27, no. 6, pp. 365–389, 2002.

[14] P. Zave, "An experiment in feature engineering," in *Programming methodology*. Springer, 2003, pp. 353–377.

[15] A. Classen, P. Heymans, and P.-Y. Schobbens, "What's in a feature: A requirements engineering perspective," in *International Conference on Fundamental Approaches to Software Engineering*. Springer, 2008, pp. 16–30.

[16] D. Batory, D. Benavides, and A. Ruiz-Cortes, "Automated analysis of feature models: challenges ahead," *Communications of the ACM*, vol. 49, no. 12, pp. 45–47, 2006.

[17] B. H. C. Cheng, P. Sawyer, N. Bencomo, and J. Whittle, "A goal-based modeling approach to develop requirements of an adaptive system with environmental uncertainty," in *Model Driven Engineering Languages and Systems*. Springer, 2009, pp. 468–483.

[18] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, no. 1, pp. 41–50, 2003.

[19] J. Yen and W. A. Tiao, "A systematic tradeoff analysis for conflicting imprecise requirements," in *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*. IEEE, 1997, pp. 87–96.

[20] J. Lee and J.-Y. Kuo, "Fuzzy decision making through trade-off analysis between criteria," *Information Sciences*, vol. 107, no. 1-4, pp. 107–126, 1998.