Exploring the Impact of Different Cost Heuristics in the Allocation of Safety Integrity Levels

Luís Silva Azevedo[1], David Parker[1], Yiannis Papadopoulos[1], Martin Walker[1], Ioannis Sorokos[1], Rui Esteves Araújo[2]

[1]Department of Computer Science, University of Hull, Hull, United Kingdom
l.p.azevedo@2012.hull.ac.uk, {d.j.parker, y.i.papadopoulos, martin.walker}@hull.ac.uk, i.sorokos@2012.hull.ac.uk
[2]INESC TEC, Faculdade de Engenharia, Universidade do Porto, Portugal
raraujo@fe.up.pt

Abstract. Contemporary safety standards prescribe processes in which system safety requirements, captured early and expressed in the form of Safety Integrity Levels (SILs), are iteratively allocated to architectural elements. Different SILs reflect different requirements stringencies and consequently different development costs. Therefore, the allocation of safety requirements is not a simple problem of applying an allocation "algebra" as treated by most standards; it is a complex optimisation problem, one of finding a strategy that minimises cost whilst meeting safety requirements. One difficulty is the lack of a commonly agreed heuristic for how costs increase between SILs. In this paper, we define this important problem; then we take the example of an automotive system and using an automated approach show that different cost heuristics lead to different optimal SIL allocations. Without automation it would have been impossible to explore the vast space of allocations and to discuss the subtleties involved in this problem.

Keywords. Dependability Analysis; Requirements Analysis; Functional Safety; SIL Allocation and Decomposition; Cost Optimisation

1 Introduction

Safety Standards, such as IEC 61508, ISO 26262, and ARP4754-A, introduce a system of classification for different levels of safety: IEC 61508 popularised the Safety Integrity Level (SIL), while ISO 26262 and ARP4754-A introduced domain-specific versions of this concept — the Automotive Safety Integrity Level (ASIL) for the automotive domain and the Development Assurance Level (DAL) for the aerospace domain. All of these serve as qualitative indicators of the required level of safety of a function or component, and generally they are broken down into 5 levels, ranging from strict requirements (e.g. SIL4, ASIL D, DAL A) to no special requirements (e.g. SIL0, QM, DAL E). These safety levels are employed as part of a top-down requirements distribution process. We focus on ISO 26262 guidelines as we will be

analysing an automotive system; however, other standards prescribe analogous rules for requirements definition, allocation and decomposition. In ISO 26262 the process of elicitation starts with a hazard and risk analysis which identifies the various malfunctions that may take place and what hazards may arise as a result. The severity, likelihood, and controllability of these hazards are then considered, and on the basis of this risk analysis, an ASIL is assigned to them; this ASIL assignment is intended to generate the necessary requirements throughout the system architecture that ensure that any risks will be decreased to an acceptable level. At this point, system level safety requirements, termed Safety Goals (SGs), are formulated, linked to system functions and inherit the ASILs of the hazard they are meant to prevent.

During subsequent development of the system, traceability to the original ASILs is maintained at all times. As the system design is refined into more detailed architectures, the original ASILs are allocated and can be decomposed throughout new subcomponents and sub-functions of the design. ISO 26262 prescribes an "ASIL algebra" to guide this process, where the various integrity levels are translated into integers (ASIL QM = 0; A = 1; B = 2; C = 3 and D = 4). The algebra is essentially an abstraction and simplification of techniques for combining probabilities under assumptions of statistical independence of failures. In this approach, components that can directly cause the corruption of a SG are assigned with the ASIL of that hazard; if, on the other hand, multiple independent components must fail together to cause a SG violation, they are allowed to share the burden of complying with the ASIL of that SG; the rationale here is that the components' total ASIL must add up to the safety level of the SG. For example, a safety level like ASIL B can be met by two independent components which each individually only meeting ASIL A (and thus effectively A + A = B).

In following the decomposition rules, there are various concerns. The first concern is raised by the complexity of modern safety-critical systems. The trend is for these architectures to become systems of systems, where multiple functions are delivered by complex networked architectural topologies and where functions can share components. The ISO standard is lacking in providing examples and detailed guidelines to support ASIL allocation in these systems. Possibly due to this lack of clarity, practitioners often make mistakes [1]. Furthermore, the safety engineer's tasks include understanding architectural failure behaviour, ensuring that component failure independence constraints are met, working efficiently through the many possible combinations for allocations, and confirming that the decomposed low-level requirements still add up to the original high-level requirements. Performing all of these manually in such complex architectures is practically impossible, and here again the standard fails to give guidance on automated support.

The second concern relates to the way SIL decomposition is being formulated in the standards as a problem solely focused on safety, where the single goal is to arrive at an allocation of integrity requirements to components of the system architecture that fulfils a set of properly elicited system level safety requirements. Naturally, standards are focused on safety, so cost implications are not really considered. On the other hand, there is no doubt that the cost implications of SIL allocation are very relevant to developers of systems. We believe there are benefits to be found if the problem of safety requirement allocation is defined in a broader way which includes costs.

Indeed, developing a component according to a given SIL, means that a set of development and validation activities needs to be undertaken. They are translated into time, efforts and in the end costs, and vary with the specific SIL prescribed to a component. Potentially, many allocation possibilities may be available, and in order to find the most advantageous, the problem needs to consider their different cost implications. To illustrate the issue, we need to turn to the very fundamentals of the techniques described in the standards. ISO 26262 gives a range of options for ASIL decomposition. For example a function with an SG that requires ASIL B can be implemented with an architecture of two components which, assuming that they fail independently, may inherit ASILs B and QM *or* A and A respectively When such options exist, cost typically provides the deciding criterion. This is precisely where the development cost differences implied by different ASILs can reveal decomposition strategies that are more cost-effective than others. During the design phase, when requirement allocation and decomposition take place, exact development costs are naturally hard to obtain. However, cost analyses can still be made on the basis of some heuristic that expresses the relative cost differences of complying with the different ASILs. For example, taking the above example, if considering a logarithmic cost increase between ASILs (ASIL QM = 0; A = 10; B = 100; C = 1000; D = 10000), when decomposing a SIL B amongst two components, C_1 and C_2, one single optimal solution is revealed.

- C_1 (ASIL QM) + C_2 (ASIL B): 0 + 100 = 100;
- **C_1 (ASIL A) + C_2 (ASIL A): 10 + 10 = 20;**
- C_1 (ASIL B) + C_2 (ASIL QM) = 100.

The example must be read carefully, not to suggest that the above is a trivial problem where decomposition opportunities can be examined independently. Components are often participating in multiple functions and numerous chains of conflicting constraints must be examined to find cost-optimal SIL allocations. This is a complex combinatorial problem where *the satisfaction of safety requirements is simply a constraint that must be met, while the real objective is the optimisation of cost.*

It is important to stress that this is not a hypothetical, but a real, important and pertinent problem. The concern about cost implications of ASILs is already evident in the automotive domain. Indeed, there has been continuous discussion within the automotive Functional Safety community to determine an appropriate ASIL cost heuristic (see for example ISO 26262 LinkedIn Forum [2]). One proposal, for instance, suggests that the cost increase between ASIL B and C is bigger than the ones from A to B and C to D. We believe that the community is aware of the importance of a plausible cost heuristic for estimating the costs of meeting safety requirements. In this paper we want to advance this discussion further by showing, with the aid of automation, that what can be seen as "optimal satisfaction" of safety requirement is in fact defined by the nature of this cost heuristic.

In section 2 we briefly outline a recently developed method for a largely automated, cost-aware, model-based allocation of safety requirements in the form of SILs. In section 3 we apply this method to a case study performed on an automotive hybrid braking system. We assume two different cost heuristics and discuss the implications

for SIL allocation. Finally in section 4 we summarise, conclude and point to future work.

2 Automatic and Optimal Decomposition of SILs

To address some of the concerns discussed above, we have recently been developing techniques [3, 4] to help automate SIL allocation and decomposition processes by extending the framework of the model-based dependability analysis tool HiP-HOPS (Hierarchically Performed Hazard Origin and Propagation Studies) [5]. HiP-HOPS is built around the concept of annotating the components of a system model with local failure logic. From these local descriptions, HiP-HOPS can synthesise fault trees for the whole system, describing how the individual component failures can propagate throughout the rest of the system and lead to hazardous conditions in the system output functionality. HiP-HOPS is aware of which components are independent by means of the assumption made in the fault propagation model and the fault tree analysis and it can automatically determine the opportunities for requirements decomposition. On the basis of this information, the tool next establishes a set of constraints based on the ASIL 'algebra' described by ISO 26262, but it can also use any analogous rules from standards affecting other industry sectors, such as the aerospace industry. Finally, HiP-HOPS initiates a search for the ASIL allocations that, while fulfilling such requirements constraints, minimize the total system's ASIL dependent costs according to some cost heuristic defined by the system designer. In related work, Mader *et al* [6] built a linear programming problem to minimize the sum of ASILs assigned to an architecture. In the aerospace sector, Bieber, Delmas and Seguin [7] used pseudo-Boolean logic to formulate the DAL decomposition problem; similarly to the work presented by Mader *et al*, the sum of DALs across the components of a system is minimized. In both cases, this can be understood as utilizing a linear fitness function to evaluate the different SIL allocation alternatives; the costs implications of each integrity level increase proportionally to the integer assigned to them by the SIL algebras. Assuming a linear cost growth is fairly simplistic and the use of other cost heuristics can be validly applied.

Our early research has shown that the number of potential combinations of allocations typically produces a vast search space. Therefore, recent work has focused on the use of metaheuristics that are known to be efficient in exploring large search spaces. Furthermore, meta-heuristics include the versatility, not present in many deterministic algorithms, to solve problems with different characteristics; this is important as we allow the system designer to input any ASIL cost function, and these have implications in the nature of the optimisation problem. Initial investigation on this has included implementations of genetic algorithms [8], a metaheuristic based on the concept of natural evolution, where a set of *candidate* solutions evolve, through *crossover* and *mutation* operations, during a fixed number of *generations* into (near) optimal solutions. More recently we have found significant improvements in solution quality and processing efficiency using a Tabu search technique [9] which is based on the work of Hansen and Lih [10]. In Tabu Search a single solution exists at any given

iteration. Specialized local search algorithms are used for that solution to travel throughout the search space together with memory mechanisms that increase diversity and grant global exploration capabilities to the algorithm. Our technique makes use of a steepest descent approach for neighbourhood exploration: the cost reductions of decrementing each of the ASILs in the current solution are analysed and the failure mode's ASIL for which this cost variation is the highest is decremented. When decrementing an ASIL means violating any decomposition constraint, a mildest ascent direction is followed by incrementing the ASIL of the failure mode which results in the lowest system cost growth. At this moment a memory mechanism forbids reverse moves for a dynamic number of iterations *p*. This is important in avoiding returning to local optima. While descending, a similar mechanism exists, prohibiting reverse moves for *p'* iterations; this allows introducing further diversity in the search by reducing the possibility of switching behaviours between solutions.

It should be noted that HiP-HOPS assigns ASILs to component failure modes instead of the components themselves. This means that there is a greater level of refinement in this method, so for example an omission of an output can be assigned a different ASIL from a commission of the output if they lead to different effects at system level. In turn, higher integrity will be required by the subcomponents that can directly or indirectly cause the omission. This feature is clearly useful for the case where a component presents more than one type of failure and therefore requirements can be more appropriately tailored in dealing with each one of them. Strictly speaking, this approach does not agree with the ISO 26262 standard, which requires that a single ASIL is allocated to a component. HiP-HOPS can easily revert to this simpler model of allocation for compatibility with the standard, assigning an ASIL to a component on the basis of its most severe failure mode. However, we believe that when requirements are allocated to subsystems which are further developed as complex networks of components, and when allocation of subsystem safety requirements to these components must be achieved, then a more refined and recursive approach where the "subsystem" can be treated as a "system" that has multiple SGs and ASIL requirements is preferable. This, we believe, could be a point worth considering in the future evolution of the ISO 26262 standard.

Finally, HiP-HOPS SIL allocation extension should be regarded as a tool to inform decision making. Ultimately, it is up to the system designer to deliberate on the list of SIL allocation possibilities supplied by the tool and make decisions on how the system architecture should evolve and how refined requirements derived from the Safety Goals are being distributed throughout the components.

In the next section we apply our automated approach to a brake-by-wire system, and in the process demonstrate the impacts of ASIL-imposed cost consideration in deriving efficient allocation strategies.

3 ASIL Allocation Cost Impacts on a Hybrid Braking System

3.1 Hybrid Brake System Description

In our work towards automating and optimising ASIL allocation we have been utilizing the model of a Hybrid Braking System (HBS) for electrical vehicles as a case study to demonstrate our advances. The model is analysed in detail in [11] and is based on a system introduced by Castro *et al* [12] where braking is achieved using the combined efforts of two actuators: In-Wheel Motors (IWMs) and Electromechanical Brakes (EMBs). IWMs decrease the kinetic energy of a vehicle transforming it into electrical energy. The latter is fed to car's powertrain batteries thus increasing driving range. However, the IWMs have limitations in regards to the amount of braking they can produce, namely at some speed regimes, or when the powertrain's batteries are close to or at a full state of charge. In that way, EMBs are used dynamically with IWMs to provide the total braking required. The HBS is a brake-by-wire system where there is no mechanical or hydraulic link between the braking pedal and the actuators. We have developed a model for the system that for illustrative purposes only considers the braking of one wheel. The model is depicted in Figure 1.

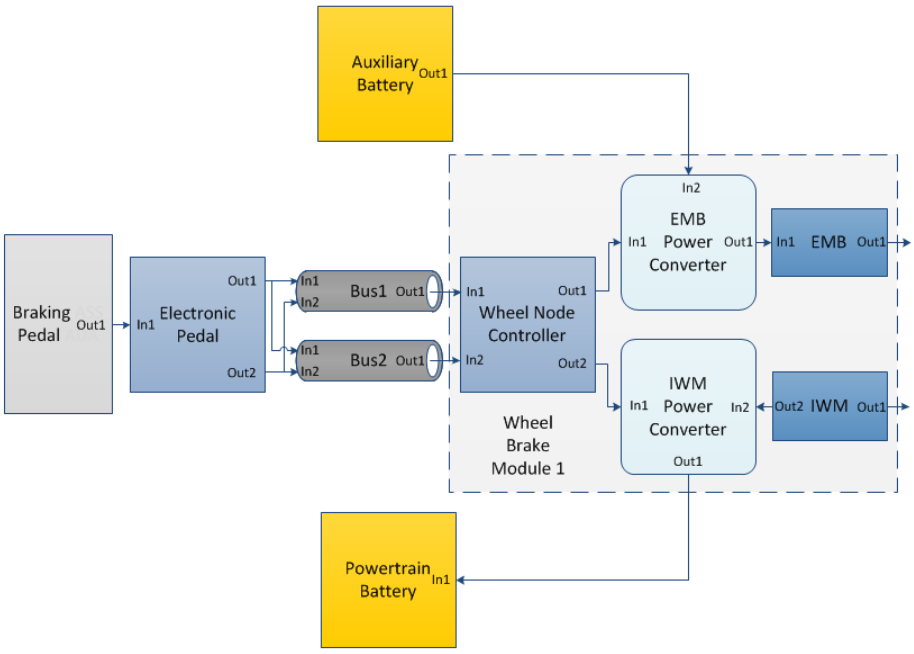

Fig. 1. Hybrid Braking System Model

For the architecture of the HBS, it is considered when the driver presses the braking pedal, his actions are sensed and processed on a redundant Electronic Pedal Unit. Braking requests are generated for each of the vehicles wheels and are sent via a duplex communications bus; these demands are received at a local Wheel Node Control-

ler (WNC) that performs calculations for the division between electromechanical and electrical braking and sends commands accordingly to the power converters that control the two types of actuators. While braking, electrical power flows from the low voltage Auxiliary Battery to the EMB; the IWM, on the other hand, acts as a generator and provides energy for the Powertrain Battery. It needs to be noted that the elements of the power architecture should be regarded as subsystems composed of multiple components. The Powertrain Battery subsystem includes, for example, a Battery Management System, and the Power Converters integrate electronic driver circuits.

For this illustrative case study we have considered two hazards: "Omission of braking" (H1) and "Braking with the wrong value" (H2). For the purpose of demonstration, H1 and H2 have been assigned with ASILs D and C, respectively. The two hazards were linked to model output deviations in the following manner:

- H1: Omission of IWM.out1 AND Omission of EMB.out1
- H2: Wrong Value of IWM.out1 OR Wrong Value of EMB.out1

We have derived failure expressions for each of the components of the HBS architecture [11]. HiP-HOPS then automatically synthesized fault trees for each of the hazards, and through their analysis unveiled:

- 1 single point of failure and 18 dual points of failure (i.e. minimal cutsets of order two) for H1;
- 10 single points of failure and 1 dual point of failure for H2.

3.2 Experimentation with Different ASIL Cost Functions

At this point we have obtained the failure information required to formulate the constraints for ASIL allocation and we now focus on the questions around using the cost functions to evaluate different ASIL allocation possibilities for the Hybrid Braking System. It is clear that there are different implications in developing the same component with each of the different ASILs. They are directly reflected in implementation and evaluation efforts, such as number of lines of code or of safety analysis execution, and consequently affect development and testing time as well as average and peak size of development and testing teams. This is deeply translated into costs, and therefore one is logically interested in finding which allocations minimize these costs across a system architecture. We have considered for this case study two functions that indicate the relative costs between compliance with the different ASILs; these serve the purpose of demonstrating the cost influence in choosing a set of ASILs for a given system design, rather than being real world applicable figures. Earlier in the paper we have mentioned that one of the potential cost heuristics presented in the ISO 26262 LinkedIn forum indicated a larger cost jump between ASILs B and C than the ones from A to B and C to D. In meeting this consideration we have formulated the *Experiential-I* cost function shown in Table 1, where the cost between B and C (20) is twice the variation between any of the other ASILs (10). We have also established a second one, *Experiential-II*, which maintains that the cost difference between ASILs B and C (15) is greater than from A to B and C to D (10). However, this func-

tion instead assumes that the cost jump between no safety considerations – ASIL QM – and ASIL A is the biggest (20).

Table 1. ASIL Cost Heuristics

Cost Heuristics / ASILs	QM	A	B	C	D
Experiential-I	0	10	20	40	50
Experiential-II	0	20	30	45	55

3.3 Cost Optimal ASIL Allocations for the Hybrid Braking System

There are 24 failure modes in the HBS and 5 different ASILs that can be allocated to each one of them. This gives a total search space size of 5^{24} ($\approx 5.96 \times 10^{16}$), which is still small enough to be finished with the exhaustive search techniques described in [3]; all optimal solutions are therefore unambiguously known. It is worth noting however, that our Tabu Search and Genetic Algorithms are able to find these allocations very efficiently. To allow discussion, we display the optimal solutions in Table 2.

Table 2. HBS Optimal ASIL Allocations for Experiential-I and Experiential-II Cost Functions

	Optimal Solutions				
	Exp-I *Cost: 390*	**Exp-II** *Cost: 585*			
Components' Failure Modes	*#1*	*#1*	*#2*	*#3*	*#4*
Braking Pedal Omission	4	4	4	4	4
Braking Pedal Value	1	1	1	1	1
Electronic Pedal Omission 1	2	4	4	4	4
Electronic Pedal Omission 2	2	0	0	0	0
Electronic Pedal Value 1	1	1	1	1	1
Electronic Pedal Value 2	0	0	0	0	0
Bus1 Omission	2	0	4	0	4
Bus2 Omission	2	4	0	4	0
WNC Omission 1	2	0	0	4	4
WNC Omission 2	2	4	4	0	0
WNC Value 1	1	1	1	1	1
WNC Value 2	1	1	1	1	1
Auxiliary Battery Omission	2	0	0	4	4
Auxiliary Battery Value	1	1	1	1	1
Powertrain Battery Omission	2	4	4	0	0
Powertrain Battery Value	1	1	1	1	1
EMB Power Converter Omission	2	0	0	4	4
EMB Power Converter Value	1	1	1	1	1
IWM Power Converter Omission	2	4	4	0	0
IWM Power Converter Value	1	1	1	1	1
EMB Omission	2	0	0	4	4
EMB Value	1	1	1	1	1
IWM Omission	2	4	4	0	0
IWM Value	1	1	1	1	1

There are obvious differences between the results of using each of the cost heuristics. The most immediate is that *Experiential-I* yields only one optimal allocation whereas *Experiential-II* yields four solutions with the same minimal cost. A closer look tells us that none of the optimal solutions of *Experiential-II* matches the one from *Experiential-I*. In this way, the choice of cost heuristic would have a definite impact on the integrity of specific components, for example, in deciding that an Omission Failure would impose development and validation measures of ASIL B *(Experiential-I*) or ASIL D/QM *(Experiential-II*) within component EMB. It is of course possible that for the HBS or another system, two different cost heuristics may work in a way that yields exactly the same optimal solution(s). However, what is important to remember is that the latter does not represent the general case, as we have demonstrated, and that the cost heuristic defines which allocation is cost-optimal. We believe this is an important realisation and should kickstart some work towards defining a plausible and widely accepted cost heuristic that could both inform automated analyses, such as the one presented in this paper, and more generally inform decisions about how to optimise allocation of safety requirements during design refinement.

3.4 Costs Refinement for More Accurate Optimal ASIL Allocations

While the results above do demonstrate the need for a unified ASIL fitness function, one can argue that it is unrealistic to consider that the efforts associated with developing a processing unit of ASIL D are even close to those required for developing a High Voltage Battery subsystem with an equal integrity level. In meeting such concerns, we have refined our approach to allow a greater granularity in costs estimation, providing the user with the ability to establish categories of components and assign relative cost weights to them. This feature is demonstrated below.

Reutilising the HBS case study, we have divided the components of its architecture in to 3 categories, as shown in Table 3. Again this was done for the sole purpose of demonstration and more accurate and meaningful divisions may be found. In the same way, we have simplified the costs of individual failure modes for illustrative purposes, and have considered that the efforts in dealing with Value and Omission failures are equal within the same component.

Table 3. HBS Components Divided in 3 Categories

Programmable Electronics	**Electronic Low Voltage**	**Electronic High Voltage**
Electronic Pedal	Auxiliary Battery	IWM
WNC	EMB Power Converter	IWM Power Converter
Communication Buses	EMB	Powertrain Battery
-	Braking Pedal	-

We have assumed the *Programmable Electronics* category is the least expensive, and have used it as the base category for relative costs definition. We have estimated that the *Electronic Low Voltage* components are 3 times more expensive than *Pro-*

grammable Electronics. It is a given that when one adds features and/or redundancy to a system, the development costs increase. That is to say, growing complexity is usually tied to an increase in risk of defect and consequently the investment in safety measures escalates. In this way, taking into account the much larger complexity usually involved in both the software and hardware elements of a high voltage architecture, *Electronic High Voltage* was assigned with the highest cost jump: 5 times the price of *Programmable Electronics*. Note that in this class we can find the main components of the traction drive system with critical expensive parts and multiple control units with embedded software. We have reapplied the two cost functions of Table 1 and have used them in conjunction with the cost weights of the 3 component categories we have devised. Our technique yielded the optimal solutions presented in Table 4.

Table 4. HBS Optimal ASIL Allocations for Experiential-I and Experiential-II Cost Functions With Cost Weights for Components Categories

	Optimal Solutions		
	Exp-I *Cost: 1030*	**Exp-II** *Cost: 1425*	
Components FM	*#1*	*#1*	*#2*
Braking Pedal Omission	4	4	4
Braking Pedal Value	1	1	1
Electronic Pedal Omission 1	2	4	4
Electronic Pedal Omission 2	2	0	0
Electronic Pedal Value 1	1	1	1
Electronic Pedal Value 2	0	0	0
Bus1 Omission	2	0	4
Bus2 Omission	2	4	0
WNC Omission 1	4	4	4
WNC Omission 2	0	0	0
WNC Value 1	1	1	1
WNC Value 2	1	1	1
Auxiliary Battery Omission	4	4	4
Auxiliary Battery Value	1	1	1
Powertrain Battery Omission	0	0	0
Powertrain Battery Value	1	1	1
EMB Power Converter Omission	4	4	4
EMB Power Converter Value	1	1	1
IWM Power Converter Omission	0	0	0
IWM Power Converter Value	1	1	1
EMB Omission	4	4	4
EMB Value	1	1	1
IWM Omission	0	0	0
IWM Value	1	1	1

It is interesting to observe that using the cost weights between components categories reveals a new optimal solution for the *Experiential-I* function, whereas for *Expe-*

riential-II the optimal solutions are a subset of the ones encountered earlier (#3 and #4 of Table 3). Even so, the minimal cost solution of *Experiential-I* is still different to the ones yielded by *Experiential-II*. Moreover, the higher cost assigned to the *Electronic High Power* category clearly biased the optimal solutions towards utilizing low ASILs for the failure modes of its components.

From the results above it was possible to demonstrate further that the use of different cost heuristics impacts the optimal allocation of ASILs, this time in the presence of relative cost differences between categories of components. Furthermore, the use of more specific cost information allowed us to reveal optimal solutions which are likely to be more accurate. For the case of the *Experiential-II* cost function, two of the solutions already identified as optimal in the previous section remained optimal following the refinement of costs. Finally, the introduction of categories and relative cost weights changes the nature of the optimisation problem; nonetheless, as in the previous step, our metaheuristics techniques were able to find the optimal solutions for this experiment, further validating their capabilities in dealing effectively with various formulations of the SIL allocation problem.

4 Conclusions

We have argued that the application of SIL allocation and decomposition is a complex combinatorial optimisation problem which must consider the optimisation of costs and not just the satisfaction of safety requirements constraints imposed by standardised SILs decomposition algebras. Through the use of a Hybrid Braking System case study we have demonstrated that such cost consideration allowed us to identify which are the most promising solutions, an effort that clearly contributes to a more efficient and cost-effective refinement of designs for safety-critical systems. Our example was equally important in revealing that different cost heuristics imply different optimality considerations; in this regard, work needs to be undertaken within each industry sector to identify plausible cost heuristics so that SIL allocation choices can be made with more confidence.

All of the deliberations above were only possible due to the use of an automated framework that enabled the exploration of the vast number of ASIL allocation solutions in our case study in the presence of different ASIL cost functions. While a definitive cost heuristic is presently not available, our method is flexible in using any that a system designer finds more suitable. Furthermore, some industries, like the automotive industry where ISO 26262 was introduced in late 2011, are still undergoing a shift from developing entire systems via application of ISO26262 towards processes in which there is reuse of Off the Shelf ASIL compliant parts. It is therefore likely that ASIL costs and their relationships might change, and it is important that our method maintains its versatility. Finally, our approach allows for designers to input costs in different levels of granularity; for example establishing categories of components with relative cost weights or even specific ASIL-dependant costs for each component. This further contributes to a more accurate determination of the best ASIL allocation strategies.

REFERENCES

1. Ward D.D. and Crozier S.E. (2012). The uses and abuses of ASIL decomposition in ISO 26262. in System Safety, incorporating the Cyber Security Conference 2012, 7th IET International Conference on. 2012.
2. Allen M. (2012).: Cost Versus ASIL. ISO 26262 Functional Safety [LinkedIn]. 2 February 2012. Available at: http://www.linkedin.com/groups/Cost-versus-ASIL-2308567.S.92692199?view=&srchtype=discussedNews&gid=2308567&item=92692199&type=member&trk=e ml-anet_dig-b_pd-ttl-cn&ut=1evtvoEm1QcBw1. [Accessed 1 May 14].
3. Papadopoulos Y., Walker M., Reiser M.-O., Weber M., Chen D., Törngren, Servat D., Abele A., Stappert F., Lönn H., Berntsson L., Johansson R., Tagliabo F., Torchiaro S., Sandberg A. (2010). Automatic Allocation of Safety Integrity Levels. In Proceedings of the 1st Workshop on Critical Automotive applications: Robustness and Safety (CARS'10), 27th April 2010, Valencia, Spain. Pages 7-10. ACM, New York, NY, USA. ISBN: 978-1-60558-915-2, doi> 10.1145/1772643.1772646
4. Azevedo L.S., Parker D., Walker M., Papadopoulos Y., and Araujo R.E. (2014) "Assisted Assignment of Automotive Safety Requirements," Software, IEEE, vol. 31, pp. 62-68, 2014.
5. Papadopoulos Y., Walker M., Parker D., Rüde E., Hamann R., Uhlig A., Grätz U., Lien R. (2011) Engineering Failure Analysis & Design Optimisation with HiP-HOPS, Journal of Engineering Failure Analysis Vol 18(2), March 2011, pages 590-608. doi> 10.1016/j.engfailanal.2010.09.025, Elsevier Science, ISSN: 1350 6307
6. Mader R., Armengaud E., Leitner A., Steger C. (2012). Automatic and Optimal Allocation of Safety Integrity Levels. In Proceedings of the Reliability and Maintainability Symposium (RAMS 2012), Reno, NV, USA. 23-26 Jan 2012, pp1-6. ISBN: 978-1-4577-1849-6, doi> 10.1109/RAMS.2012.6175431
7. Bieber, P., Delmas, R., and Seguin, C. (2011). DALculus: theory and tool for development assurance level allocation. In S. B. Francesco Flammini (Ed.), 30th international conference on Computer safety, reliability, and security (SAFECOMP'11), Naples, Italy. Pages 43-56. Springer-Verlag, Berlin, Heidelberg.
8. Parker D., Walker M., Azevedo L., Papadopoulos Y., Araujo R. (2013) Automatic Decomposition and Allocation of Safety Integrity Levels using a Penalty-based Genetic Algorithm. Proceedings of the 26th International Conference on Industrial, Engineering, and other Applications of Applied Intelligent Systems (IEA/AIE 2012): Special session on Decision Support for Safety-Related Systems. 17-21st June, Amsterdam, The Netherlands.
9. Azevedo L.S., Parker D., Walker M., Papadopoulos Y., and Araujo R. E. (2013) Automatic Decomposition of Safety Integrity Levels: Optimisation by Tabu Search. 2nd Workshop on Critical Automotive applications: Robustness & Safety (CARS), at the 32nd International Conference on Computer Safety, Reliability, and Security (SAFECOMP'13), Toulouse, France, 2013.
10. Hansen, P. and Lih, K.-W (1996) Heuristic reliability optimization by tabu search. Annals of Operations Research, volume (63), pp. 321-336
11. Azevedo L.P. (2012) Hybrid Braking System for Electrical Vehicles: Functional Safety, M.Sc. thesis, Dept. Elect. Eng., Porto Univ., Porto, Portugal, 2012.
12. de Castro R., Araújo R.E., and Freitas D.. (2011) "Hybrid ABS with Electric motor and friction Brakes," presented at the IAVSD2011 - 22nd International Symposium on Dynamics of Vehicles on Roads and Tracks, Manchester, UK, 2011.