10th World Congress on Structural and Multidisciplinary Optimization
May 19 - 24, 2013, Orlando, Florida, USA

Decomposition-based analysis of the multi-physical coupling structure in LED System-in-Package design

E.C.M. de Borst[1,2], L.F.P. Etman[1], A.W.J. Gielen[2], A.T. Hofkamp[1], and J.E. Rooda[1]

[1] Eindhoven University of Technology, The Netherlands, [2] TNO Eindhoven, The Netherlands, email:
e.c.m.d.borst@student.tue.nl, l.f.p.etman@tue.nl, sander.gielen@tno.nl, a.t.hofkamp@tue.nl, j.e.rooda@tue.nl

1. Abstract
LED System-in-Package aims to reduce manufacturing costs of LED lighting products through integration of components into one single package, based on semiconductor technology. This causes tight multi-physical coupling which complicates the design process. We present a method to analyze the coupling structure of the LED SiP. A specification language is used to specify the input-output (binary) relationships between design variables, physical responses, design objectives and constraints. An adapted design structure matrix (DSM), representing the linkage, is automatically generated from the specification. The rows and columns of the adapted DSM matrix are subsequently re-ordered using partitioning and sequencing algorithms to provide insight in the coupling structure. A LED SiP prototype is used as illustration.

2. Keywords: Decomposition, multi-physical coupling, partitioning, sequencing, matrix representation

3. Introduction
LED System-in-Package is a novel lighting concept that aims to reduce the cost of the next generation LED lighting products. The LED SiP integrates LED chips with driver chips, sensors, radio-frequency interfaces and possibly other components into a single semiconductor solution, see Figure 1. The integrated and miniaturized package reduces manufacturing and material costs. However, due to the integration, the product design becomes tightly coupled [1]. Unlike traditional devices built on printed-circuit-boards, the interaction between the different functional components goes beyond the electrical domain: heat generation, electromagnetic fields, optical interference, and mechanical loads between the functional components must be taken into account to obtain a LED SiP with a good performance and lifetime. In this study we analyze the multi-physical coupling structure combining techniques from multi-disciplinary design optimization and product architecture design.

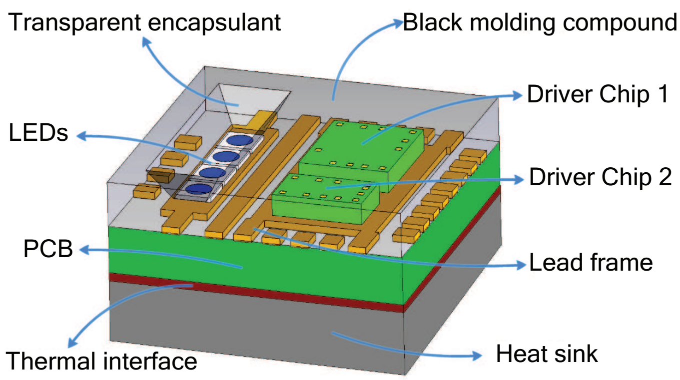

Figure 1: LED SiP design concept

During the LED SiP product design, mechanical, thermal, electrical and optical aspects have to be taken into account. In particular thermal management plays a keyrole, since the performance and lifetime of components is heavily affected by high temperatures within the package. For example, the actual light output of high brightness LEDs depends on the junction temperature in the LED. As temperature rises the light output decreases. Also the quality of the light deteriorates under increasing temperatures, since LEDs tend to shift wavelength at higher temperatures, resulting in off-color light output. On a longer time horizon, due to high temperatures and high electrical currents, driver and lenses tend to degrade resulting in lumen depreciation and color shift. The LED SiP can also fail catastrophically, when one of the components in the LED SiP instantly fails. Such catastrophic failures may be electrically or thermo-mechanically induced.

Typical design decisions for the LED SiP are associated with, amongst others, the type, dimensions and mutual placement of the various components that make up the SiP, the routing of interconnects, and the package. The main design objectives are: performance, lifetime, and costs. The performance is described by the luminous efficacy, i.e. the amount of light (lm) per amount of electrical energy (W), and the quality of the light output. The lifetime of the LED SiP is a function of failure due to lumen depreciation and catastrophic failure of components. The costs typically relate to material and manufacturing costs.

Several teams of engineers, typically from different companies, will be involved during the design of a LED SiP product. Each team has a specialization in one particular discipline or physical building block. In product design of existing LED lighting systems, the design of the electrical circuit is usually leading, followed by the design of the components and the design of the package. Ultimately proper functioning of the full system is analyzed. However, the coupled nature due to the SiP concept complicates prediction of the system behavior and thus the process of decision making. The implications of component design decisions on system behavior are not fully understood. A design change in one part of the design may result in an unexpected change in behavior in another part.

Various methods to analyze interactions in systems exist in the fields of engineering design, product management, product design and multidisciplinary design optimization [2]. Commonly matrix and graph representations are used. For instance, the design structure matrix (DSM) [3, 4] displays the relationships between the elements of a system in a square matrix. Another well-known matrix representation is the functional dependence table (FDT) [5], which is often used in multidisciplinary design optimization. The FDT displays the dependence of objective and constraint functions on the design variables. To identify coupling structure, the DSM and the FDT can be partitioned using partitioning algorithms. By re-ordering rows and columns, blocks of strongly coupled subproblems are identified. In case of the DSM, directionality of the interactions may be used to re-order matrix elements such that a sequence with minimal feedback is obtained.

Our research aims at the development of methods to analyze the coupling structure of the physical behavior of the LED SiP. Our focus is on the various disciplinary and component responses that play a role. A "classical" FDT representation expresses the objective and constraint functions directly in terms of the decision variables, which fails to provide the necessary insights regarding the interactions between the responses. We wish to include the notion of physical responses in the matrix representation by introducing an intermediate response level of abstraction in the description of the functional relationships between design variables and design objectives/constraints. Furthermore, we wish to be able to identify the input-output relations to reveal directionality of the "information flow".

In our approach we combine the DSM concept (to show the mutual interactions between elements) and the FDT concept (to relate design variables to design objectives) into a square matrix representing the interactions between design variables, responses, and design objectives/constraints. This matrix is similar to the adapted DSM matrix of Allison *et al.* [6], but we use it for a different purpose. A second ingredient that we use is the Ψ language [7] to specify the functional relations between design variables, responses and objectives, and then automatically generate the adapted DSM matrix. The matrix is subsequently partitioned using the Graclus graph clustering algorithm [8] to minimize a weighted criterion of partition sizes and partition interactions. Also a combination of an exhaustive sequencing strategy and a Dulmage-Mendelsohn decomposition algorithm [9] is employed to arrive at a matrix with minimal feedback-coupling. The re-ordered matrix serves as a means to analyze the SiP coupling structure, and to compare with current design decomposition practice in LED lighting product development.

The paper is organized as follows. In the sequel section the DSM and the FDT are further explained, as well the purpose of partitioning and sequencing these matrices. Section 5 presents our proposed method. In Section 6 this is applied to a test case based on a recently manufactured LED SiP prototype. Section 7 presents some concluding remarks.

4. Matrix representations

4.1. Design Structure Matrix (DSM)

The Design Structure Matrix (DSM) originates from Steward [4], who proposed a matrix-based technique to manage the design of complex systems, in particular regarding the interactions between elements of

	A	B	C	D	E	F	G
A						X	
B	X		X	X			X
C				X			X
D		X	X		X		X
E	X			X		X	
F	X				X		
G		X	X	X			

(a) Design structure matrix

	x_1	x_2	x_3	x_4	x_5	x_6
f_1	X			X		
f_2		X			X	
h_1	X	X				
h_2	X	X	X			
h_3				X		X
h_4	X			X		
h_5		X			X	
h_6			X			X

(b) Functional dependence table

Figure 2: Matrix representations

the system. The main purpose of a DSM is to illuminate structure and aid in the design of products, processes and organizations [10, 3].

A DSM is a square $N \times N$ matrix with identical row and column labels, representing the elements of the modeled system. These elements can be, amongst others, system components, parameters, tasks, and activities. An off-diagonal mark, which may be a "1" or an "X" mark, represents linkage between two elements. The main-diagonal elements of the DSM are usually shaded. Besides binary marks, also weighted interactions can be presented in a "numerical" DSM, for example to differentiate between strong and weak dependencies. Figure 2a shows an example of a DSM, representing the interactions between seven elements by means of cross-marks.

The DSM may be symmetrical or asymmetrical. In a symmetrical DSM directionality of interactions is not considered. In an asymmetrical DSM, the input-output direction of the linkage is taken into account. In this regard, two conventions exist. In the DSM convention due to Steward, a cross at row i, column j means that element j is input to element i. However, the opposite convention, element i is input to element j, is also frequently used [3]. In this paper we adopt Steward's convention.

4.2. Functional Dependence Table (FDT)

The functional dependence table (FDT), is an $m \times n$ representation matrix aimed at the decomposition of model-based design optimization problems [5, 11]. Figure 2b shows the FDT. Each row of the FDT represents a design function, typically an objective function or a constraint function. Each column represents a design variable. Table entry (i, j) is non-empty if the i-th function depends on the j-th variable.

4.3 Partitioning

Partitioning aims to find groups of strongly coupled elements. In the context of the DSM partitioning means parts of the design that should be considered together (an illustration is given in Figure 3); in the context of the FDT this means parts of the optimization problem that should be optimized together. In partitioning there are two competing goals: minimizing the size of partitions, and minimizing the number of interactions between partitions. Partitioning algorithms re-order the rows and columns in the matrix, to minimize a weighted criterion of partition size and partition interaction. Two classes of partitioning algorithms exist: algorithms that operate directly on the matrix, and algorithms that use a graph representation of the coupled system (noting that the matrix and graph representation are closely related)

4.4. Sequencing

Sequencing is the reordering of the DSM's rows and columns such that the new DSM arrangement has a minimum of feedback marks (output of an element that is input to an element with a lower index). A system without feedback coupling can be sequentially designed, starting with the first element, then the second element, etcetera. Feedback coupling implies design iterations. Using Steward's convention, feedback marks are located above the matrix diagonal. Thus the goal of sequencing is to minimize the number of interactions marks above the matrix diagonal, i.e. transform the DSM into a lower block triangular matrix form. A frequently used method to permute the rows and columns of a matrix to

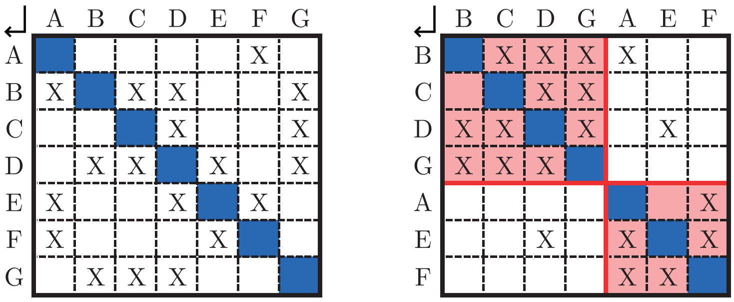

Figure 3: Partitioning of a DSM

arrive at the lower block triangular matrix form is the Dulmage-Mendelsohn decomposition [9]. Figure 4 illustrates the sequencing of a DSM.

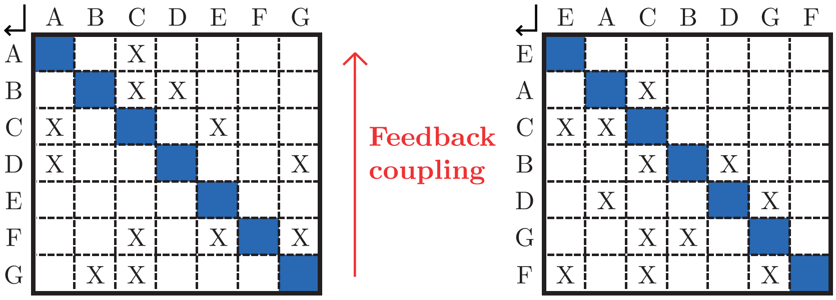

Figure 4: Sequencing of a DSM

5. Proposed method

In order to relate the design variables x to the design objectives f, response functions r are introduced. $\mathbf{r} = r(\mathbf{x}, \mathbf{y})$ denotes the set of output response variables of response function r. The inputs of a response function are design variables $\mathbf{x}$ and response output from other response functions $\mathbf{y}$. Figure 5a illustrates the coupling between design variables $\mathbf{x}$, responses $\mathbf{r}$, and design objectives $\mathbf{f}$. The response functions represent the physical relations that have to be considered to model the multidisciplinary behavior of the various components and their interactions. Our main interest is to identify which input-output relations are present. The actual functional relations are not quantified. We assume a binary representation matrix.

5.1. Representation matrix

The coupling between design variables $\mathbf{x}$, responses $\mathbf{r}$, and design objectives $\mathbf{f}$ is represented using the adapted DSM as shown in Figure 5b. Note that the rows corresponding to the design variables are all empty. This is due to the fact that design variables are always input to other elements. Similarly, the columns corresponding to the objective functions are empty (objectives only have input).

In industrial applications the size of the matrix may grow to hundreds or even thousands of elements. In the DSM literature, the identification of interactions between elements is mainly based on design documentation and interviews with designers. This interaction modeling is not straightforward, and is quite difficult both for larger systems, and in the early design phase, see [12, 13, 14, 15]. A similar difficulty is present for the identification of the relations between design variables $\mathbf{x}$, responses $\mathbf{r}$, and objectives $\mathbf{f}$.

5.2. Specifying interactions

We observe however that we do not need to manually fill the matrix, checking entry by entry of the matrix. Instead, we propose to use a language to specify the variables and their interactions, and subsequently generate the matrix automatically. We have adopted the Ψ specification language [7] to demonstrate the advantage of such an approach. The Ψ specification language was originally intended as a linguistic software tool for specification of partitioned problems in decomposition-based design optimization. The

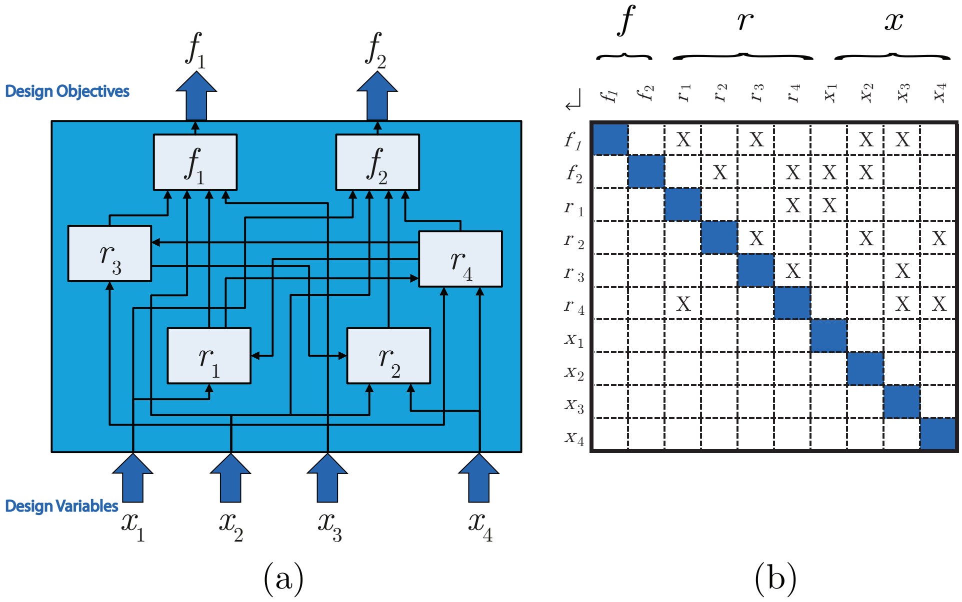

Figure 5: Illustration of the representation of the coupling structure using response functions

language elements include design variables, objective functions, constraint functions, as well as response functions. The language provides an intuitive environment for compact specification of partitioned optimization problems. Subproblems can be defined, and subsequently assembled into larger subsystems and systems. This assembly feature presents a significant advantage for specifying the interactions in the LED SiP application: the specification of variables and their interactions can be carried out locally for subsystems of the LED SiP; the coupling between subsystems can be specified at a higher system level.

Listing 1: Example Ψ specification of a LED chip

```
comp LED_Thermal =
|[ extvar  Tj, Ploss, Td, Tl, Ta
   intvar  Rled
   resfunc Tj = a1(Ploss, Td, Tl, Ta, Rled)
]|

comp LED_Optical =
|[ extvar  If, Tj
   intvar  LEDdeg
   objfunc F1(If,Tj, LEDdeg)
   resfunc LEDdeg = a2(If,Tj)
]|

comp LED_Electrical =
|[ extvar  Ploss, Tj, If, Id, Vd
   resfunc If  = a3(Id,Vd,Tj), Ploss = a4(Id,Vd)
]|

syst LED =
|[ sub LT: LED_Thermal, LO: LED_Optical, LE: LED_Electrical
   link    LT.Tj -- LO.Tj, LT.Tj -- LE.Tj, LE.If -- LO.If
       ,   LE.Ploss -- LT.Ploss
   alias   Id = LE.Id, Vd = LE.Vd, Td = LT.Td, Tj = LT.Tj, Tl = LT.Tl
       ,   Ta = LT.Ta
]|
```

Listing 1 shows an example of a simplified Ψ specification for a LED chip in the SiP. The system behavior of the LED comprises three disciplinary aspects, which are electrical, thermal, and optical behavior. Therefore, the (sub)system `LED` consists of thee sub-components `Thermal`, `Optical`, and `Electrical`. Each component contains functions and variables. The functions may be response functions (`resfunc`), objective functions (`objfunc`) and constraint functions (`confunc`). For example, response variable T_j, the LED junction temperature, follows from response function a_1 with as input arguments the heat loss of the LED $Ploss$, the temperatures of adjacent physical components, driver temperature T_d, lead frame temperature T_l, ambient temperature T_a, and the thermal resistance R_{led}. R_{led} is a design variable which occurs only locally in this component, and is therefore specified as an internal variable (`intvar`). The external variables (`extvar`) are variables that are coupled to other components in the Ψ specification.

Within the `syst` environment external variables of various components can be linked. `sub` denotes which components are part of the system defined by `syst`. The linkage is specified using the `link` statement. For example, the LED junction temperature T_j is linked between the thermal and the optical component, by `LT.Tj -- LO.Tj`. In turn, the `syst LED` may be a subsystem of a larger system. Variables may again be linked between the LED system and other systems in the specification. To pass variables through, the `alias` statement facilitates shorthand notation.

5.3. Partitioning and sequencing

The adapted DSM is automatically generated from the Ψ specification. Subsequently, structure can be identified by re-ordering of rows and columns.

To this end, we first apply a partitioning algorithm, to group together design variables, responses, objectives, and constraints that are strongly coupled. Due to the anticipated size of the matrix (hundreds of elements), probably only a graph-based partitioning algorithm is applicable. We opted for Graclus [8], since it is able to compute partitions of unequal size (unlike Metis [16] for instance). However, Graclus is intended for undirected graphs, while our adapted DSM includes direction. This direction information (i.e. the asymmetry of the matrix) is accounted for by introducing edge weights to the graph (an edge is the connection between two nodes). A connection in one direction has weight 1; otherwise the connection has weight 2. A limitation of Graclus is that it requires as input the desired number of partitions, and only minimizes for the number of interactions between partitions. Graclus determines which design variables, responses, and objectives belong to each partition.

Secondly, sequencing is carried out without affecting the partitioning outcome. To do so, we start with an exhaustive search to find the optimal sequence of the partitions such that the amount of feedback coupling between partitions is minimized (here we assume that the number of partitions is sufficiently small to allow for such enumeration procedure). Then, Dulmage-Mendelsohn decomposition is applied to each partition separately (i.e. within the partition), in order to obtain the optimal sequence of elements within the partitions.

5.4. Procedure

A summary of the proposed procedure is as follows:

1. **Specify the x, r, f interactions using the Ψ language**
2. **Generate the adapted DSM containing the x, r, f interactions**
3. **Partition the matrix to find the strongly coupled parts**
4. **Find an ordering with minimal number of feedback interactions, by sequencing the partitions, and sequencing within each partition**

Figure 6 gives an illustration of the outcome of the procedure for the sample problem presented in Figure 5b. Variables x_2, x_4, r_2, x_1, and f_2 are grouped together in partition 1; variables x_3, r_1, r_4, r_3, f_1 are grouped together in partition 2. Note that if we express f_1 and f_2 directly in terms of x_1 through x_4, we observe that f_1 as well as f_2 depend on all design variables, leaving no room to find structure based on an FDT only.

6. LED SiP design case

A LED SiP prototype developed by TNO and partners is used as a case to validate the proposed four step procedure. We assume an early phase of the design, where design decisions have not yet been made,

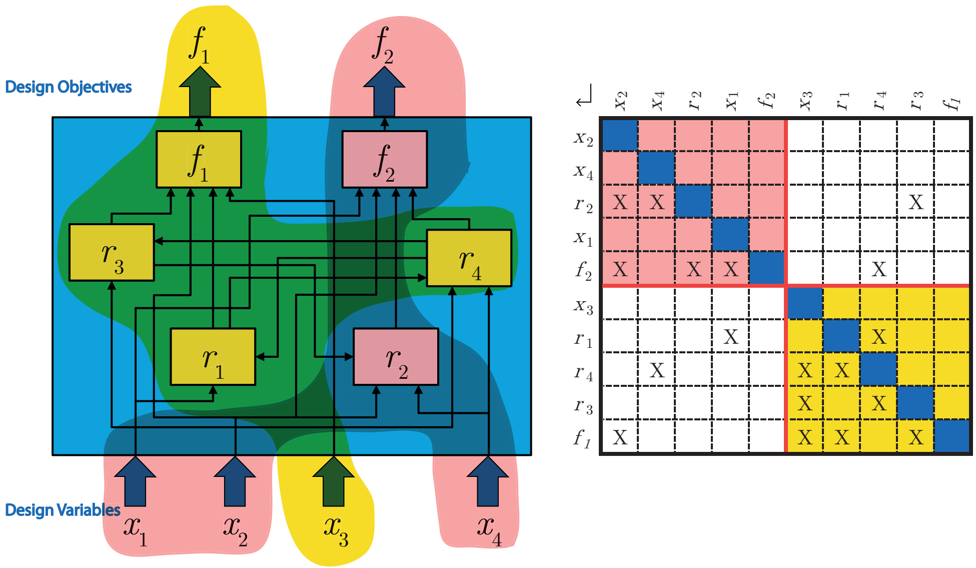

Figure 6: Illustration of decomposition based on the system's coupling structure

without prejudice of the current LED SiP design prototype. The system elements and their interactions were obtained by means of a literature review and interviews with designers. The literature provided background knowledge regarding the physical working principles of the various components. The interviews provided information on the functionality, components, and disciplinary aspects that should be considered to design a LED SiP. Furthermore, the interviews revealed that the design engineers prefer to first decompose the LED SiP into components, and subsequently decompose each component regarding the relevant disciplinary aspects. Also, there was general agreement that thermal and mechanical disciplinary analysis is necessary not only at the component level, but also at the level of the complete LED SiP design, to guarantee proper functioning.

The preference by the engineers to first consider components and subsequently consider disciplines is used as framework to build the Ψ specification of the LED SiP. In the Ψ specification, LED dies, driver chips, lead frame, transparent encapsulant, phosphor film, black compound, interconnects, I/O pads, die attach, heat sink, and PCB components are distinguished. For each of these subsystems, corresponding thermal, optical, electrical and mechanical disciplinary components are specified. The behavior is specified by means of variables, response functions, constraint functions and objective functions. Additionally, a thermal and a mechanical analysis system are included representing system wide analysis. Grand total, the LED Sip specification consists of 45 Ψ (sub)systems, where each system may contain one or more disciplinary components. The systems are linked together by means of linkage of variables.

From the Ψ specification the adapted DSM containing the interactions between design variables, responses, objectives and constraints was automatically generated. The generated matrix has size 711*711. Among the 711 elements are 450 design variables, 253 responses, 6 constraints and 2 design objectives.

Next, graph partitioning using Graclus is applied. Since the desired number of partitions is unknown, the adapted DSM matrix is partitioned for a range of partition numbers, from 2 to approximately 700 partitions. The following cost function, due to [17], is used to evaluate the trade-off between the number of interactions between partitions and the size of the partitions:

$$C_{total} = \underbrace{\sum (A(i,j) + A(j,i)) \cdot S_{partition(y)}}_{\text{Interactions within partitions}} + \underbrace{\sum (A(i,j) + A(j,i)) \cdot S_A}_{\text{Interactions outside partitions}} \tag{1}$$

The first summation accounts for the interactions that are within partitions, by adding the size of the corresponding partition y to the costs for each interaction. The second summation takes the interactions that are outside partitioning into account, by adding the size of the full matrix to the costs for each interaction. Figure 7 shows the resulting calculated costs for the range of partitioning solutions. The cost value first decreases as the number of partitions increases, which is caused by the decrease in partition size. Then the cost value increases again, as a result of an increasing number of interactions outside the partitions. The minimum value for this cost function is found around 10 partitions. Note that by changing the weights between the two parts of the cost function the minimum changes to the left or to the right.

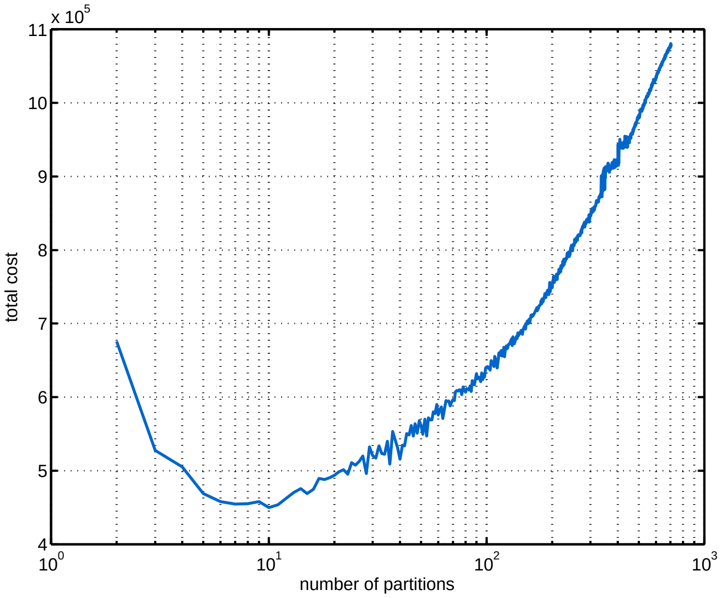

Figure 7: Calculated cost function

Figure 8 shows the adapted DSM for the solution with 10 partitions after additional sequencing. Each partition represents a strongly coupled part of the design. The thermal analysis partition and the mechanical analysis partition are both large partitions and are placed early in the sequence. As a consequence the current design practice to start from the electrical aspects of the design may be less suited, since the main electrical part is placed in sequence after the thermal and mechanical part. Apparently first the mechanical and thermal design should be considered, before designers should consider the electrical and optical aspects.

7. Concluding remarks

An adapted design structure matrix (DSM) defining the input-output coupling between design variables, physical responses, objective functions and constraint functions, is suitable to analyse the multi-physical coupling structure in LED SiP design. The introduction of the responses plays a key role in the representation of the coupling structure. To generate the DSM, the use of a specification language to specify the linkages has proven to be advantageous. By partitioning and sequencing of the DSM the coupling structure can be revealed. For the LED SiP prototype we observed that thermal and mechanical system analysis should play a more prominent role in guiding the design process of the SiP.

8. References

[1] E. C. M. de Borst, A. W. J. Gielen, and L. F. P. Etman, "Coupling structure in LED System-in-Package design: A physical responses-based critical parameter sheet like approach," in *4th Electronics System Integration Technologies Conference (ESTC 2012), Amsterdam*, IEEE, September 2012.

[2] A. Kusiak and N. Larson, "Decomposition and representation methods in mechanical design," *Journal of Mechanical Design*, vol. 117, p. 17, 1995.

[3] S. D. Eppinger and T. R. Browning, *Design Structure Matrix Methods and Applications*. Engineering Systems, Mit Press, 2012.

[4] D. V. Steward, "The design structure system: A method for managing the design of complex systems," *IEEE Transactions on Engineering Management*, vol. EM-28, no. 3, pp. 71–74, 1981.

[5] T. C. Wagner, *A general decomposition methodology for optimal system design*. PhD thesis, University of Michigan., 1993.

[6] J. T. Allison, M. Kokkolaras, and P. Y. Papalambros, "Optimal partitioning and coordination decisions in decomposition-based design optimization," *Journal of Mechanical Design*, vol. 131, no. 8, p. 081008, 2009.

[7] S. Tosserams, A. T. Hofkamp, L. F. P. Etman, and J. E. Rooda, "A specification language for problem partitioning in decomposition-based design optimization," *Structural and Multidisciplinary Optimization*, vol. 42, no. 5, pp. 707–723, 2010.

[8] I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors a multilevel approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 1944–1957, 2007.

[9] A. L. Dulmage and N. S. Mendelsohn, "Coverings of bipartite graphs," *Canadian Journal of Mathematics*, vol. 10, pp. 517–534, 1958.

[10] T. R. Browning, "Applying the design structure matrix to system decomposition and integration problems: a review and new directions," *IEEE Transactions on Engineering Management*, vol. 48, no. 3, pp. 292–306, 2001.

[11] N. F. Michelena and P. Y. Papalambros, "A hypergraph framework for optimal model-based decomposition of design problems," *Computational Optimization and Applications*, vol. 8, no. 2, pp. 173–196, 1997.

[12] A. H. Tilstra, C. C. Seepersad, and K. L. Wood, "The repeatability of high definition design structure matrix (hddsm) models for representing product architecture," in *Proceedings of the ASME 2010 International Design Engineering Technical Conferences, Quebec*, ASME, August 2010.

[13] Q. Dong, *Predicting and managing system interactions at early phase of the product development process*. PhD thesis, Massachusetts Institute of Technology, 2002.

[14] F. Steffen, D. C. Schmitz, and P. Wieland Biedermann, "Improving data quality in dsm modelling: A structural comparison approach," *Proceedings of the 18th International Conference on Engineering Design (ICED11), Vol. 4*, pp. 369–380, 2011.

[15] A. H. Tilstra, *Representing Product Architecture and Analyzing Evolvable Design Characteristics*. PhD thesis, University of Texas, 2010.

[16] G. Karypis, "Metis-unstructured graph partitioning and sparse matrix ordering system, version 5.0," 2011.

[17] R. E. Thebeau and D. E. Whitney, *Knowledge management of system interfaces and interactions from product development processes*. PhD thesis, Massachusetts Institute of Technology, 2001.

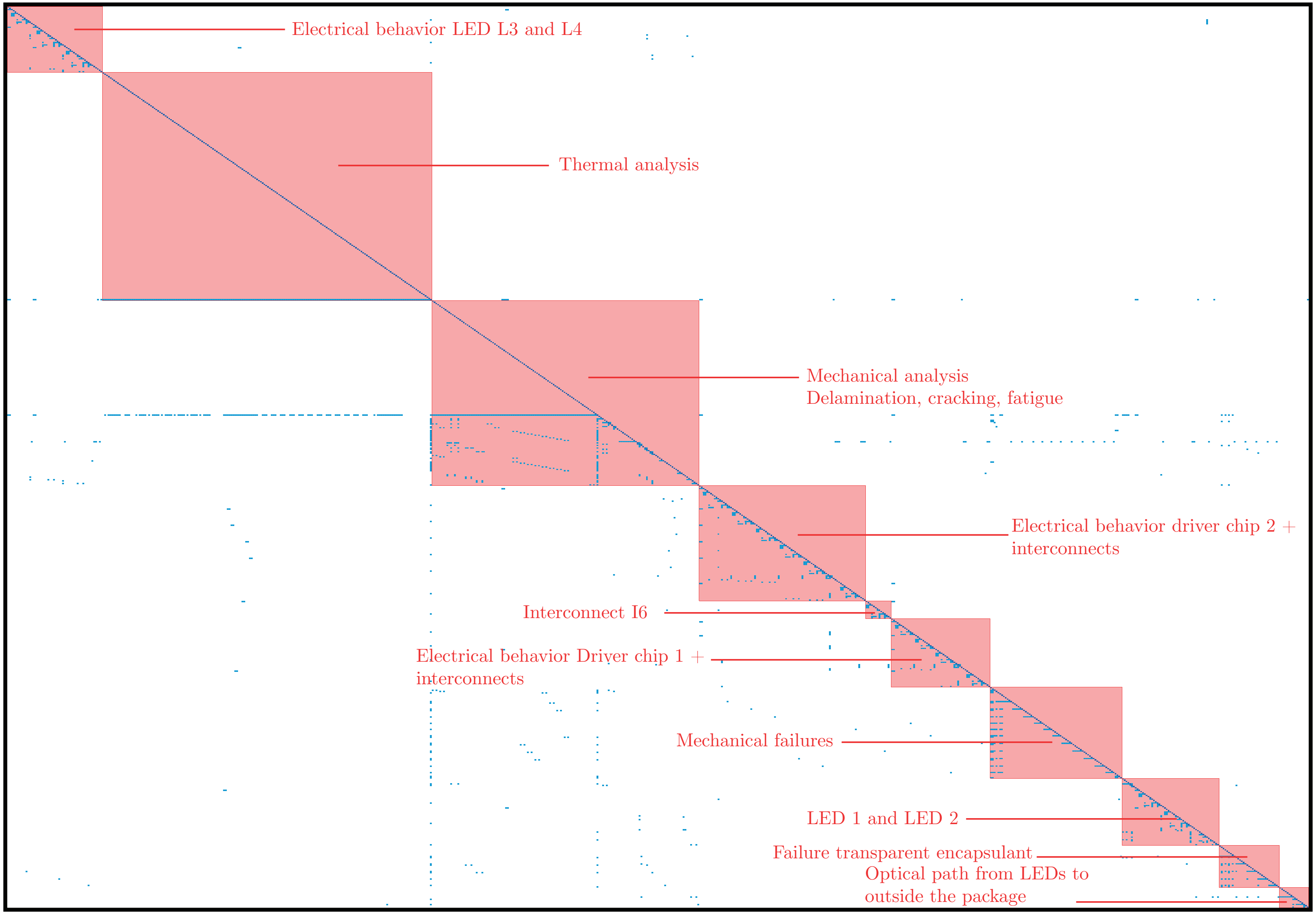

Figure 8: Representation matrix of the LED SiP design case partitioned into 10 partitions