Data Partitioning via High Dimensional Model Representation by Using Paralel Computing

ENGİN KANAL
İstanbul Technical University
Informatics Institute
Maslak, İstanbul
TÜRKİYE (TURKEY)

METİN DEMİRALP
İstanbul Technical University
Informatics Institute
Maslak, İstanbul
TÜRKİYE (TURKEY)

Abstract: The multivariate functions become plague of plethora when the number of their arguments increases to high values as much as 2 digits numbers, because of the limitations of today's computers. Instead of computer programming directly, the mathematically efficient multivariate function representations must be developed before attempting computer programming. One of those methods is "High Dimensional Model Representation (HDMR)"Even when these types of methods are applied on the multivariate functions, the solution cost of the modified problem after the application of the method is still complicated for a stand–alone computer. In that case, the parallel programming helps us. Our main goal of this work is to modify the method for parallel computing.

Key–Words: Parallel Programming, Data Partitioning, Multivariate Approximation, High Dimensional Model Representation

1 Introduction

When a natural event is analyzed, the number of factors affecting the event is higher than calculated. General way to overcome this is to eliminate or ignore some of factors which will be less effective. However as the investigated event complicates, the number of affecting factors increases and including all these factors, within nowaday computer technology, can not be suitable to the calculation limitations on this types of problems. "High Dimensionel Model Representation (HDMR)"is perhaps the most fruitful solution to those multidimensional problems.

With the help of data partitioning via HDMR, a given N–variate function can be approximated. Even the approximation is achieved mathematically, computational cost may still exceeds the limitations for one PC. Therefore, the method must be modified for parallel programming to apply on real–world problems.

In this work, we offer an idea to modify the HDMR functions for parallel programming. For this purpose, we investigate the mathematical structure of data partitoning via HDMR method. Then we use the structure to modify the method. Although we have no any parallel implementation yet and this presentation is rather conceptual, we explain the next steps of this work in the future works section.

The paper is organized as follows. We begin with a brief introduction of HDMR. Then we mention data partitioning via HDMR. We offer the method that modifies the data partitioning via HDMR method for parallel programming in main section. We discuss future works at the last section.

2 HDMR

The equation for High Dimensional Model Representation for a given multivariate function is as follows according to Sobol Theory.

$$\begin{aligned} f(x_1,...,x_N) &= f_0+\sum_{i_1=1}^{N} f_{i_1}(x_{i_1}) \\ &+ \sum_{i_1,i_2=1 i_1<i_2}^{N} f_{i_1i_2}(x_{i_1},x_{i_2}) \\ &+ \cdots + f_{12\ldots N}(x_1,...,x_N) \end{aligned} \tag{1}$$

This expansion is a finite sum and composed of a constant term, univariate terms, bivariate terms and so on up to the N–variate term. These are the HDMR components of the given multivariate function. For this moment, the important step is to define the functions on the right hand side of the equation (1).

$$\int_0^1 dx_s f_{i_1,i_2,\ldots i_k}(x_{i_1},\ldots,x_{i_k}) = 0$$

$$s = i_1, \ldots, i_k \tag{2}$$

These functions must satisfy orthogonality condition via an inner product.

$$\begin{aligned}&(f_{i_1 i_2 \ldots i_k}, f_{i_1 i_2 \ldots i_l}) = 0,\\ &\{i_1, i_2, \ldots, i_k\} \not\equiv \{i_1, i_2, \ldots, i_l\},\ \ 1 \le k, l \le N\end{aligned} \tag{3}$$

According to the equality defined above and orthogonality condition, to obtain the constant term, f_0, the equation (1) is integrated over all variables as follows.

$$\int_0^1 \cdots \int_0^1 dx_1 \ldots dx_N f(x_1, \ldots, x_N) = f_0 \tag{4}$$

Similarly, integrating the equation (1) on all variables except x_i give the below equality and then univariate term $f_i(x_i)$.

$$\begin{aligned}&\underbrace{\int_0^1 \cdots \int_0^1}_{N-1\ times} dx_1 \ldots dx_{i-1} dx_{i+1} \ldots dx_N\\ &f(x_1, \ldots, x_N) = f_0 + f_i(x_i)\ \ \forall i = 1, \ldots, N\end{aligned} \tag{5}$$

This method is improved and extended by the Rabitz Group and applied most of reseach areas [2, 3, 4, 5]. Instead of the unit weight functions and the integration interval [0,1] in Sobol Theory, Rabitz Group refer to use the equation below.

$$\begin{aligned}&\int_{a_1}^{b_1} dx_1 \cdots \int_{a_N}^{b_N} dx_N W(x_1, ..., x_N) f_i(x_i) = 0,\\ &1 \le i \le N\end{aligned} \tag{6}$$

The weight function $W(x_1, ..., x_N)$ is defined as follows.

$$\begin{aligned}W(x_1, \ldots, x_N) &\equiv \prod_{j=1}^{N} W_j(x_j),\\ x_j \in [\, a_j\, ,\, b_j\,], &\qquad 1 \le j \le N\end{aligned} \tag{7}$$

Additionally, the functions must satisfy the normalization condition below.

$$f_0 = \int_{a_j}^{b_j} dx_j W_j(x_j) = 1, \qquad 1 \le j \le N \tag{8}$$

Similarly to Sobol's, the HDMR functions are obtained on $W(x_1, ..., x_N)$ weight function as follows.

$$\begin{aligned}f_0 \ \ = &\int_{a_1}^{b_1} dx_1 \cdots \int_{a_1}^{b_1} dx_N W(x_1, ..., x_N)\\ &\times f(x_1, \ldots, x_N)\end{aligned} \tag{9}$$

$$\begin{aligned}f_k(x_k) = &\int_{a_1}^{b_1} dx_1 W_1(x_1)\\ &\cdots \int_{a_{k-1}}^{b_{k-1}} dx_{k-1} W_{k-1}(x_{k-1})\\ &\times \int_{a_{k+1}}^{b_{k+1}} dx_{k+1} W_{k+1}(x_{k+1}) \cdots\\ &\int_{a_N}^{b_N} dx_N W_N(x_N)\\ &\left[f_0 + \sum_{i_1=1}^{N} f_{i_1}(x_{i_1}) + \cdots \right] - f_0,\\ &1 \le k \le N\end{aligned} \tag{10}$$

$$\begin{aligned}f_{k_1 k_2}(x_1, ..., x_N) = &\int_{a_1}^{b_1} dx_1 W_1(x_1)\\ &\cdots \int_{a_{k_1-1}}^{b_{k_1-1}} dx_{k_1-1} W_{k_1-1}(x_{k_1-1})\\ &\times \int_{a_{k_1+1}}^{b_{k_1+1}} dx_{k_1+1} W_{k_1+1}(x_{k_1+1})\\ &\cdots \int_{a_{k_2-1}}^{b_{k_2-1}} dx_{k_2-1} W_{k_2-1}(x_{k_2-1})\\ &\times \int_{a_{k_2+1}}^{b_{k_2+1}} dx_{k_2+1} W_{k_2+1}(x_{k_2+1})\\ &\cdots \int_{a_N}^{b_N} dx_N W_N(x_N)\\ &\times \left[f_0 + \sum_{i_1=1}^{N} f_{i_1}(x_{i_1}) + \cdots \right]\\ &- f_{k_1}(x_{k_1}) - f_{k_2}(x_{k_2}) - f_0,\\ &1 \le k_1 < k_2 \le N\end{aligned} \tag{11}$$

3 Data Partitioning

Since we need to perform a multivariate interpolation on finite number of discrete points we can extend the domain of HDMR variables to entire space without imposing any extra condition. Hence, we assume that the interval of the independent variables is $(-\infty, \infty)$. It is considered that, the structure of the function, $f(x_1, ..., x_N)$, is not given analytically. Instead it is specified as the values on finite number of points in the cartesian space defined by the independent variables. These points are defined through a cartesian product. For this definition, first the data of the variable x_j is defined as the following set.

$$\mathcal{D}_j \equiv \left\{ \xi_j^{(k_j)} \right\}_{k_j=1}^{k_j=n_j} = \left\{ \xi_j^{(1)}, ..., \xi_j^{(n_j)} \right\},\ \ 1 \le j \le N \tag{12}$$

The cartesian product mentioned above can be constructed from these sets as follows.

$$\mathcal{D} \equiv \mathcal{D}_1 \times \mathcal{D}_2 \times \cdots \times \mathcal{D}_N \tag{13}$$

Explicit set theoretical definition of $\mathcal{D}$ can be given as

$$\mathcal{D} \equiv \{\tau | \tau = (x_1, x_2, ..., x_N), x_j \in \mathcal{D}_j, 1 \leq j \leq N\} \tag{14}$$

$\mathcal{D}$ consists of N-tuples, that is, points in the N dimensional cartesian space. The structure which needs to be created through the interpolation must include values of the function $f(x_1, ..., x_N)$ on these points only. This structure can be obtained by formatting the weight function for this purpose. In this sense the necessary action is to define the weight function as a linear combination of several Dirac delta functions. Hence, the following weight function can be selected.

$$W_j(x_j) \equiv \sum_{k_j=1}^{n_j} \alpha_{k_j}^{(j)} \delta\left(x_j - \xi_j^{(k_j)}\right), \quad x_j \in [a_j, b_j], 1 \leq j \leq N \tag{15}$$

The left hand side of the equation describing the normalization criterion (8) can be expressed for the weight function mentioned above as follows

$$\int_{a_j}^{b_j} dx_j \sum_{k_j=1}^{n_j} \alpha_{k_j}^{(j)} \delta\left(x_j - \xi_j^{(k_j)}\right) = \sum_{k_j=1}^{n_j} \alpha_{k_j}^{(j)}, \quad 1 \leq j \leq N \tag{16}$$

because of the properties of the Dirac delta function. This enables us to obtain the following relation as a condition on the linear combination coefficients of Dirac delta functions.

$$\sum_{k_j=1}^{n_j} \alpha_{k_j}^{(j)} = 1, \qquad 1 \leq j \leq N \tag{17}$$

Using the relations in (9) and (15) and the properties mentioned above, the following equality is obtained to determine constant term,

$$\begin{aligned} f_0 = & \int_{a_1}^{b_1} dx_1 \sum_{k_1=1}^{n_1} \alpha_{k_1}^{(1)} \delta\left(x_1 - \xi_1^{(k_1)}\right) \cdots \times \\ & \times \int_{a_N}^{b_N} dx_N \sum_{k_N=1}^{n_N} \alpha_{k_N}^{(N)} \delta\left(x_N - \xi_N^{(k_N)}\right) \\ & f(x_1, ..., x_N) \end{aligned} \tag{18}$$

With the help of defined weight function, we can say

$$f_0 = \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \cdots \sum_{k_N=1}^{n_N} \left(\prod_{i_1=1}^{N} \alpha_{k_{i_1}}^{(i_1)} \right) \times f\left(\xi_1^{(k_1)}, ..., \xi_N^{(k_N)}\right) \tag{19}$$

This equality can be rewritten as follows.

$$f_0 \equiv \sum_{\tau \in \mathcal{D}} \zeta(\tau) f(\tau) \tag{20}$$

where

$$\tau = \left(\xi_1^{(k_1)}, ..., \xi_N^{(k_N)}\right), \ \zeta(\tau) = \alpha_1^{(k_1)} \cdots \alpha_N^{(k_N)}, \quad 1 \leq k_j \leq n_j, \quad 1 \leq j \leq N \tag{21}$$

We can use the definition (10) to determine HDMR univariate components $f_m(x_m)$.

$$\begin{aligned} & f_m\left(\xi_m^{(k_m)}\right) = \\ & \int_{a_1}^{b_1} dx_1 W(x_1) \cdots \int_{a_{m-1}}^{b_{m-1}} dx_{m-1} W(x_{m-1}) \\ & \times \int_{a_{m+1}}^{b_{m+1}} dx_{m+1} W(x_{m+1}) \\ & \cdots \int_{a_N}^{b_N} dx_N W(x_N) f(x_1, ..., x_N) - f_0 \end{aligned} \tag{22}$$

When the right hand side of the above relation is rewritten by taking the univariate factors of the weight function into consideration, the following $N-1$ fold integral structure is obtained.

$$\begin{aligned} f_m\left(\xi_m^{(k_m)}\right) = & \int_{a_1}^{b_1} dx_1 \sum_{k_1=1}^{n_1} \alpha_{k_1}^{(1)} \delta\left(x_1 - \xi_1^{(k_1)}\right) \cdots \\ & \times \int_{a_{m-1}}^{b_{m-1}} dx_{m-1} \sum_{k_{m-1}=1}^{n_{m-1}} \alpha_{k_{m-1}}^{(m-1)} \delta\left(x_{m-1} - \xi_{m-1}^{(k_{m-1})}\right) \\ & \times \int_{a_{m+1}}^{b_{m+1}} dx_{m+1} \sum_{k_{m+1}=1}^{n_{m+1}} \alpha_{k_{m+1}}^{(m+1)} \delta\left(x_{m+1} - \xi_{m+1}^{(k_{m+1})}\right) \\ & \cdots \int_{a_N}^{b_N} dx_N \sum_{k_N=1}^{n_N} \alpha_{k_N}^{(N)} \delta\left(x_N - \xi_N^{(k_N)}\right) \\ & \times f(x_1, ..., x_N) - f_0 \end{aligned} \tag{23}$$

Evaluating the above integrals and inserting the explicit form of the constant term given in (20), these relations can be rewritten in the following more general form.

$$\begin{aligned} f_m\left(\xi_m^{(k_m)}\right) = & \sum_{\tau_m \in \mathcal{D}^{(m)}} \zeta_m(\tau_m) f(\tau_m, \xi_m^{(k_m)}) \\ & - \sum_{\tau \in \mathcal{D}} \zeta(\tau) f(\tau) \end{aligned} \tag{24}$$

where

$$\mathcal{D}^{(m)} \equiv \{\tau_m | \tau_m (x_1, ..., x_{m-1}, x_{m+1}, ..., x_N)$$

$$
\begin{aligned}
& , \quad x_j \in \mathcal{D}_j, 1 \le j \le N, j \ne m\}, \\
\tau_m &= \left(\xi_1^{(k_1)}, ..., \xi_{m-1}^{(k_{m-1})}, \xi_{m+1}^{(k_{m+1})}, ..., \xi_N^{(k_N)}\right), \\
\zeta_m(\tau_m) &= \alpha_1^{(k_1)} \cdots \alpha_{m-1}^{(k_{m-1})} \alpha_{m+1}^{(k_{m+1})} \cdots \alpha_N^{(k_N)}, \\
\xi_m^{(k_m)} &\in \mathcal{D}_m, 1 \le k_m \le n_m, 1 \le m \le N \quad (25)
\end{aligned}
$$

With the equality above, N tables of ordered pairs such that m^{th} table constains n_m $(1 \le m \le N)$ ordered pairs of $f_m(x_m)$ are obtained.

4 Modifying The HDMR Data Partitioning Algorithm for Parallel Programming

In this section, we focus on a small data and explain the idea on that data. For this purpose, we take the data as follows.

$$
\mathcal{D}_1 = \left\{\xi_1^{(1)}, \xi_1^{(2)}, \xi_1^{(3)}\right\}, \ \mathcal{D}_2 = \left\{\xi_2^{(1)}, \xi_2^{(2)}\right\}, \ \mathcal{D}_3 = \left\{\xi_3^{(1)}, \xi_3^{(2)}\right\} \quad (26)
$$

Assume that we have all the $f(x_1, x_2, x_3)$ values. First, we discuss modifying the about equality (19) for the HDMR constant component, f_0. It is easy to realize that the equality uses all of the given values, and nothing else because of the weight functions, for calculating the constant term as follows.

$$
\begin{aligned}
f_0 &= (\alpha_1^{(1)}\alpha_1^{(2)}\alpha_1^{(3)}) f(\xi_1^{(1)}, \xi_2^{(1)}, \xi_3^{(1)}) \\
&+ (\alpha_1^{(1)}\alpha_1^{(2)}\alpha_2^{(3)}) f(\xi_1^{(1)}, \xi_2^{(1)}, \xi_3^{(2)}) \\
&+ (\alpha_1^{(1)}\alpha_2^{(2)}\alpha_1^{(3)}) f(\xi_1^{(1)}, \xi_2^{(2)}, \xi_3^{(1)}) \\
&+ (\alpha_1^{(1)}\alpha_2^{(2)}\alpha_2^{(3)}) f(\xi_1^{(1)}, \xi_2^{(2)}, \xi_3^{(2)}) \\
&+ (\alpha_2^{(1)}\alpha_1^{(2)}\alpha_1^{(3)}) f(\xi_1^{(2)}, \xi_2^{(1)}, \xi_3^{(1)}) \\
&+ (\alpha_2^{(1)}\alpha_1^{(2)}\alpha_2^{(3)}) f(\xi_1^{(2)}, \xi_2^{(1)}, \xi_3^{(2)}) \\
&+ (\alpha_2^{(1)}\alpha_2^{(2)}\alpha_1^{(3)}) f(\xi_1^{(2)}, \xi_2^{(2)}, \xi_3^{(1)}) \\
&+ (\alpha_2^{(1)}\alpha_2^{(2)}\alpha_2^{(3)}) f(\xi_1^{(2)}, \xi_2^{(2)}, \xi_3^{(2)}) \\
&+ (\alpha_3^{(1)}\alpha_1^{(2)}\alpha_1^{(3)}) f(\xi_1^{(3)}, \xi_2^{(1)}, \xi_3^{(1)}) \\
&+ (\alpha_3^{(1)}\alpha_1^{(2)}\alpha_2^{(3)}) f(\xi_1^{(3)}, \xi_2^{(1)}, \xi_3^{(2)}) \\
&+ (\alpha_3^{(1)}\alpha_2^{(2)}\alpha_1^{(3)}) f(\xi_1^{(3)}, \xi_2^{(2)}, \xi_3^{(1)}) \\
&+ (\alpha_3^{(1)}\alpha_2^{(2)}\alpha_2^{(3)}) f(\xi_1^{(3)}, \xi_2^{(2)}, \xi_3^{(2)}) \quad (27)
\end{aligned}
$$

The equality means the cartesian product of $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$ sets. The mathematical structure of the equality can be explained via "Graph Cartesian Product"for

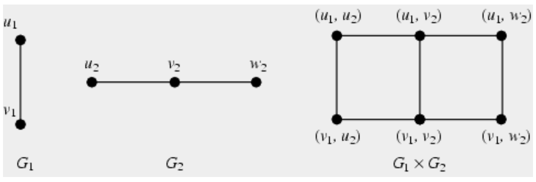


Figure 1: Graph Cartesian Product

$G_1(v_1, u_1)$ and $G_2(u_2, v_2, w_2)$ as follows. For our data, graph cartesian product would be a 3-D cube. At this point we need one definition for explaining the equality (19) as a graph. The definition is the k-Partite Graph. A k-partite graph is a graph whose graph vertices can be partitioned into k disjoint sets so that no two vertices within the same set are adjacent [8, 9]. The data sets we have are disjoint, so we can use this definition. Additionally, every trio of (x_1, x_2, x_3) of the graph vertices in the 3 sets are adjacent. k-partite graph also is a complete 3-partite graph such that every pair of graph vertices in the 3 sets are adjacent. Therefore, the elements of the equality (20) turns out to be the following graphs.

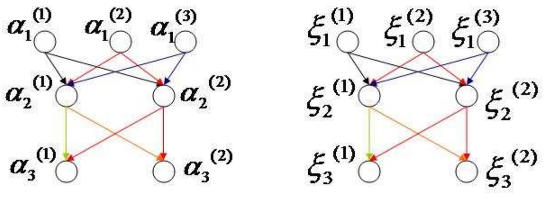


Figure 2: Complete Partite Graph

After this, we need one more step to express the graph as an array. This array will be an input array for the parallel algorithm. For this step, we use "Graph–Structured Stack"definition. A graph–structured stack is a directed acyclic graph where each directed path is a stack [10]. In the following diagrams, there are four stacks: $\{7, 3, 1, 0\}$, $\{7, 4, 1, 0\}$, $\{7, 5, 2, 0\}$, and $\{8, 6, 2, 0\}$.

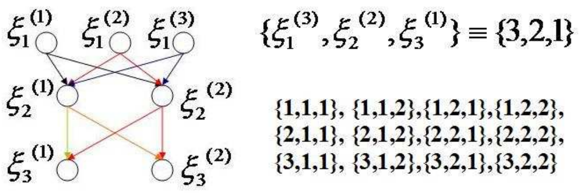


Figure 3: Graph–Structured Stack

With the help of this definition, we can express all (x_1, x_2, x_3) trios used in (20) as an array as follows.

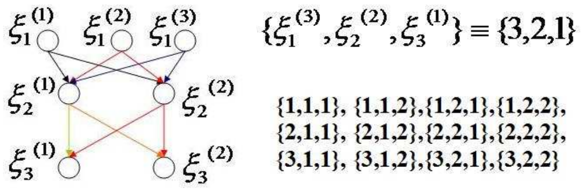


Figure 4: Graph–Strucured Stack for Cartesian Product of $\mathcal{D}_1$, $\mathcal{D}_2$ an $\mathcal{D}_3$

Where the elements in every row on the left hand side of Figure 4 are in $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$ sets, respectively. At the top right side of the figure, there is an abbreviation for the trio (x_1, x_2, x_3) and at the bottom right there is the cartesian product of $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$ as an array with the help of the definition, Graph-Strucured Stack. Although there is no problem to obtain the HDMR constant term f_0 for parallel programming, because the process uses all of the data, calculating the HDMR univariate terms $f_i(x_i)$ has a different nature. The difference is the fact that each calculation process use different part of the data. For example, there is a cartesian product of the three sets also, in the calculation process of $f_1(\xi_1^{(1)})$. Howver this time we assume that the set $\mathcal{D}_1$ consists only the element $\xi_1^{(1)}$, so the other combinations connect $\xi_2^{(1)}$ and $\xi_3^{(1)}$ are ignored. The scheme for using input data to calculate univariate term $f_1(\xi_1^{(1)})$ is given below where on the left side of the figure the graph-

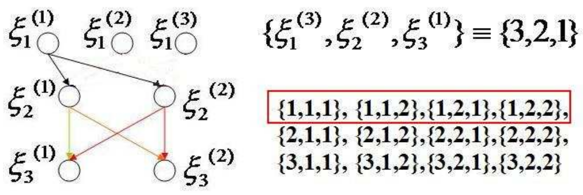
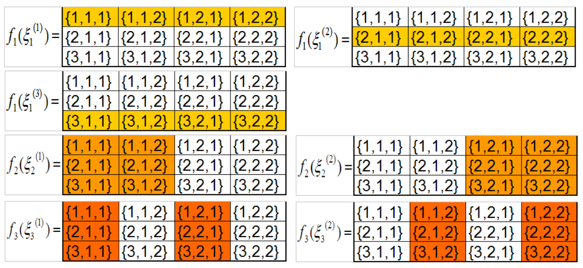


Figure 5: Scheme for using input data to calculate $f_1(\xi_1^{(1)})$

structured stack appears for $f_1(\xi_1^{(1)})$ and on the right side there is marked part of data used for the calculation of $f_1(\xi_1^{(1)})$. With the aid of the strategy for preparing input data of the algorithm, using the part of data depends on a scheme. Using the data parts scheme while calculating the HDMR univariate terms is as follows.



$$f_1(\xi_1^{(1)}) = \begin{array}{|c|c|c|c|}\hline \{1,1,1\} & \{1,1,2\} & \{1,2,1\} & \{1,2,2\} \\ \hline \{2,1,1\} & \{2,1,2\} & \{2,2,1\} & \{2,2,2\} \\ \hline \{3,1,1\} & \{3,1,2\} & \{3,2,1\} & \{3,2,2\} \\ \hline \end{array} \quad f_1(\xi_1^{(2)}) = \begin{array}{|c|c|c|c|}\hline \{1,1,1\} & \{1,1,2\} & \{1,2,1\} & \{1,2,2\} \\ \hline \{2,1,1\} & \{2,1,2\} & \{2,2,1\} & \{2,2,2\} \\ \hline \{3,1,1\} & \{3,1,2\} & \{3,2,1\} & \{3,2,2\} \\ \hline \end{array}$$

$$f_1(\xi_1^{(3)}) = \begin{array}{|c|c|c|c|}\hline \{1,1,1\} & \{1,1,2\} & \{1,2,1\} & \{1,2,2\} \\ \hline \{2,1,1\} & \{2,1,2\} & \{2,2,1\} & \{2,2,2\} \\ \hline \{3,1,1\} & \{3,1,2\} & \{3,2,1\} & \{3,2,2\} \\ \hline \end{array}$$

$$f_2(\xi_2^{(1)}) = \begin{array}{|c|c|c|c|}\hline \{1,1,1\} & \{1,1,2\} & \{1,2,1\} & \{1,2,2\} \\ \hline \{2,1,1\} & \{2,1,2\} & \{2,2,1\} & \{2,2,2\} \\ \hline \{3,1,1\} & \{3,1,2\} & \{3,2,1\} & \{3,2,2\} \\ \hline \end{array} \quad f_2(\xi_2^{(2)}) = \begin{array}{|c|c|c|c|}\hline \{1,1,1\} & \{1,1,2\} & \{1,2,1\} & \{1,2,2\} \\ \hline \{2,1,1\} & \{2,1,2\} & \{2,2,1\} & \{2,2,2\} \\ \hline \{3,1,1\} & \{3,1,2\} & \{3,2,1\} & \{3,2,2\} \\ \hline \end{array}$$

$$f_3(\xi_3^{(1)}) = \begin{array}{|c|c|c|c|}\hline \{1,1,1\} & \{1,1,2\} & \{1,2,1\} & \{1,2,2\} \\ \hline \{2,1,1\} & \{2,1,2\} & \{2,2,1\} & \{2,2,2\} \\ \hline \{3,1,1\} & \{3,1,2\} & \{3,2,1\} & \{3,2,2\} \\ \hline \end{array} \quad f_3(\xi_3^{(2)}) = \begin{array}{|c|c|c|c|}\hline \{1,1,1\} & \{1,1,2\} & \{1,2,1\} & \{1,2,2\} \\ \hline \{2,1,1\} & \{2,1,2\} & \{2,2,1\} & \{2,2,2\} \\ \hline \{3,1,1\} & \{3,1,2\} & \{3,2,1\} & \{3,2,2\} \\ \hline \end{array}$$

Figure 6: Scheme for using input data to calculate univariate terms

where the array is shown as a matrix to facilitate easy grasping the issue.

5 Concluding Remarks

The next step after this conceptual design is the parallel programming of the algorithm. After this step, the efficiency of the parallel program will be analyzed and applied on the real–data and reported in our future publications.

Acknowledgements: The second author is grateful to Turkish Academy of Sciences and both authors thank to WSEAS, for their supports.

References:

[1] **Sobol, I. M., and**, 1993, Sensitivity Estimates for Nonlinear Mathematical Models, *Mathematical Modelling and Computational Ex pe riments*, **1**, pp. 407-414.

[2] **Rabitz, H., and Alış, Ö. F.**, 1999, Additive and Multiplicate High Dimensional Representation General Foundations of High Dimensional Model Representations, *J. Math. Chem.*, **25**, pp. 197-233.

[3] **Alış, Ö. F., and Rabitz, H.**, 2001, Efficient Implementation of High Dimensional Model Representations, *J. Math. Chem.*, **29**, pp. 127-142.

[4] **Li, G., Rosenthal, C., and Rabitz, H.**, 2001, High Dimensional Model Representations, *J. Phys. Chem. A*, **105**, pp. 7765-7777.

[5] **Li, G., Wang, S. W., Rosenthal, C., and Rabitz, H.**, 2001, High Dimensional Model Representations Generated from Low Dimensional Data Samples. I. mp-Cut-YBMG, *J. Math. Chem.*, **30**, pp. 1-30.

[6] **M.Alper Tunga**, 2006, Data Partitioning and Multivariate Interpolation via Various High Dimensional Model Representation, *Ph.D Thesis*

[7] **WolframMathWorld**, 2007, Graph Cartesian Product, *http://mathworld.wolfram.com/ GraphCartesianProduct.html*

[8] **Mohar Bojan, Pisanski Tomaz and White Arthur T.**, 1990, Embeddings of Cartesian Products of Nearly Bipartite Graphs, *J. Graph Theory*, **14**, pp. 301-310..

[9] **Sergei L. Bezrukov, Robert Elsasser**, 2003, Edge–Isoperimetric Problems for Cartesian Powers of Regular Graphs, *Theoretical Computer Science*, **307**, pp. 473 - 492

[10] **www.answers.com**, 2007, Graph-Structured Stack, *http://www.answers.com/topic/graphstru cturedstack*

[11] **WolframMathWorld**, 2007, Forest, *http://math world.wolfram.com/Forest.html*

[12] **Demiralp M.**, 2002, Çarpımsallaştırılmış Yüksek Boyutlu Model Gösterilimi; Gösterilim Çarpanlarının Belirlenmesi, *12. Ulusal Mekanik Kongresi Bildirileri, (10-14 Eylül 2001, Konya)*, pp. 261-268.

[13] **Tunga A., Demiralp M.**, 2002, A High Dimensional Model Representation Based Approximation Scheme for Multivariate Interpolation, *Proceedings GIS 2002, (International Symposium on Geographic Information Systems, September 23-26, 2002, İstanbul)*, pp. 678-686.

[14] **Tunga A., Demiralp M., Yanalak M.**, 2002, A High Dimensional Model Representation Based on Lagrange Interpolation for Digital Elevation Models, *Proceedings GIS 2002, (International Symposium on Geographic Information Systems, September 23-26, 2002 İstanbul)*, pp. 687-696.

[15] **Tunga A., Demiralp M**, 2004, A Factorized High Dimensional Model Representation on the Partitioned Random Discrete Data, *ANACM*, **1**, pp. 231-241.

[16] **Tunga, M. A, Demiralp M.**, 2005, A Factorized High Dimensional Model Representation on the Nodes of a Finite Hyperprismatic Regular Grid, *Applied Mathematics and Computation*, **164**, pp. 865-883.

[17] **Tunga, M. A, Demiralp M.**, 2006, Hybrid High Dimensional Model Representation (HHDMR) on the Partitioned Data, *Journal of Computational and Applied Mathematics*, **185**, pp. 107-132.

[18] **Li G., Artamonov M., Rabitz H., Wang SW, Georgopoulos P. G., Demiralp M.**, 2002, High-Dimensional Model Representations Generated from Low Order Terms-lp-RS-HDMR, *Journal of Computational Chemistry*, **24**, pp. 647-656.

[19] **Demiralp M.**, 2003, High Dimensional Model Representation and its Applications, *Tools For Mathematical Modelling*, **9**, pp. 146-159.

[20] **Demiralp M.**, 2006, Monotonously Increasing Multiplicativity Measurers in Factorized High Dimensional Model Representation, *ICNAAM-2006 Proceedings, (International Conference on Numerical Analysis and Applied Mathematics, -1519 September, 2006, Hersonissos, Crete, G-reece).* pp 109-112.

[21] **Demiralp M.**, 2006, Transformational High Dimensional Model Representation, *Lecture Series on Computer and Computational Sciences, Recent Progress in Computational Science and Engineering, Selected Papers from the International Conference of Computational Methods in Sciences and Engineering 2006 (ICCMSE 20-06), 27 October-1 November, 2006, Chania, C-rete, Greece*. **7A**. pp 128-131.

[22] **Demiralp M.**, 2006, Weight Parameters Optimization to Get Maximum Constancy in High Dimensional Model Representation, *WSEAS T-ransaction on Mathematics*, **5**, pp. 1177-1181.

[23] **Demiralp M.**, 2006, Illustrative Implementations to Show How Logarithm Based High Dimensional Model Representation Works for Various Function Structures, *WSEAS Transaction on Computers*, **5**, pp. 1333-1338.

[24] **Demiralp M.**, 2006, Plain and Logarithmic High Dimensional Model Representation and -the Effect on Their Types on Univariance Level, 2006, *WSEAS Transaction on Mathematics*, **5**, pp. 582-588.