Optimizing the Design Structure Matrix for improving the Supply Chain

AbdoulRahman M.AlShaar[1] , Hisham M.AbdelSalam[2]

[1]Datascience Company, Mokatam, Cairo, Egypt

[2]Faculty of Computers and Information, Cairo University, Cairo, Egypt.

ABSTRACT

Weak links always represent the bottleneck of any coordination of the flows of information, materials and functions across the supply chain to achieve higher organizational performance. Since Supply Chain is a complex system that involve many other systems, integration and communication we simplified it using Design Structure Matrix (DSM) tool to ease of representing and clustering related elements. The key of success is coordinating them and better understand systems behaviors. This paper optimized DSM using Evolutionary Algorithms Genetic Algorithm with discrete problem representation for two clusters, three clusters and the dynamic number of clusters.

Keywords: Design Structure Matrix, Genetic Algorithm, Particle Swarm Optimization, Supply Chain

1. INTRODUCTION

A supply chain can be defined as network of facilities and distribution options which performs the functions of materials procurement and transformation into intermediate and finished products, till the distribution of these finished products to customers. Supply chains used in both service and manufacturing organizations. Realistic supply chains have multiple end products with shared components, facilities and capacities [1]. Supply chain takes attention and calls for serious researches, the key of success for any manufactory or company is to find the best way to meet ever-rising customer expectation at a manageable cost. Minimizing interference between system elements in supply chain is always the key of competition in producing best products with best design in lower cost, to cover complex system interactions for various functional area, and to enhance the effectiveness of communication and coordination among different system elements and its flow which make it a most to use effective tool such as Design Structure matrix DSM.

In today's highly changeable markets, it is critical for every enterprise to dynamically maintain its Supply chain efficiency and flexibility, optimizing decisions in Supply Chain has been increasingly recognized as a fateful competitive factor, industry and companies should integrate and coordinate all business operations starting from pre-process of ordering the raw material and going through product distribution to retaining strengthen in competitive market with sustainability considerations [2]. Sustainability being involved the objectives of social, economic, resources and environmental sustainability; some of them are conflicting [3]. Supply chain from academic and specially industry become very attractive to the chemical and electromechanical industries [4], [5], [6]. Design and optimization of supply chain configuration is considered as strategic level planning starting from the highest level.

Supply chain needs information and communication flow to be clear for building strategic businesses which have main goal of coordinating operations across departments. Guiding us to Design Structure matrix "DSM" which was introduced by Steward (1981) [7] in the beginning to analyze the process of engineering design. DSM can be take advantage of solving types of problems in various kinds of projects which is based on the descriptions of task flows using graphic symbols in addition its suitability for sequencing management, and modeling information flow with ability of coping with design issues of complex systems [8].It can be a suitable tool that's help in building clear view by managing coordinating of operation and facilitating best information flow and communication [8]. It is quantifying the information flow of activity factors handling new product development and employing using clustering methods. Utilizing the proposed partition and tearing algorithm to reorder matrices. It is ultimately help designers and managers grabbing and viewing interactive information flow between each of the activity factors to help them planning an optimum design process [9].

Developing supply chain systematic depends on identifying and quantifying the interactions between system elements, decomposing large interdependent group of system elements into smaller and managing sub-groups to improve the structure of the supply chain system. DSM which is useful tool that is suitable for engineers merely deals with those three main problems [10]. Using DSM in project scheduling incapable to deal with time factors of information exchange because it is difficult to determine during the planning stage of a project even by estimating the time factors of information exchange based previous projects in same region, experts facing difficulty to find a precise time, especially for irregular projects such as NPD (new product development), software project managing or organizational change.

This paper optimized the systematic approach used in [10] to reach best solution to minimize the interaction among related system elements calculated using Numerical interaction density to evaluate the cluster performance, Cluster analysis is used and optimized to decompose large interdependent group into smaller one in order to provide a better supply chain system structure. An illustrative example in the paper is used to prove the optimization of the system structure. The paper main objective is to increase the communication between system elements leading to improve the system structure of a supply chain by decreasing the interference between each other. Evolutionary algorithms Discrete Genetic Algorithm and Particle Swarm Optimization had been used to cluster system elements.

2. PROBLEM DESCRIPTION

Supply Chain researchers used to focus on better clustering the system elements, our proposed implementation is working on interdependent functions then on the aggregated interaction strength, using Genetic Algorithm. The GA was used to optimize the aggregated matrix of the three function's interactions Information, Material and functional functions then optimizing them in giving two clusters, three clusters and finally the dynamic cluster size optimized by applying the discrete search and take into consideration penalty added if any of constraints been violated.
Engineers used DSM (Design Structure Matrix) to analyze the complex system in Supply Chain as it is the best tool to build a clear view of the system as well as the interdependence between system elements initially it introduced by Steward [7] for analyzing engineering design process which is equivalent to an adjacent matrix, which dividing the system into sub systems in an $n * n$ matrix and a value indicating the interaction strength is represented in a relation between elements If there exists an information flow from element i to element j, the value of element ij will have the strength value as illustrated in table 1 There are three basic types of relationships among design tasks: sequential (dependent), parallel (independent),and coupled (interdependent).

For example, in Figure. 1(a), task A provides information to task B with strength of 0.2, i.e., their relationship is sequential. For tasks B and C, these two tasks are independent since no information flow exists between them. There is information exchange between tasks D and E, and these two tasks are regarded as coupled with strength of 0.7 and 0.4. As shown in Table. 1(a), there are marks above the diagonal. Such an above-diagonal mark represents information feedback from a downstream task. Hence, the initial execution of the upstream task is based on some conjecture of the input information. If the conjecture proves to be inaccurate, the upstream task has to be re-executed. To avoid such reworks, the original DSM shall be partitioned to re-order the columns and rows so as to eliminate or reduce the feedback marks. Figure. 1(b) shows a partitioned DSM of the matrix in Figure. 1(a). After partitioning, the number of feedbacks is reduced and three clusters are observed. The binary DSM provides a basic form for representing design information flows.

To deal with complexity of reality design projects, various extensions have been proposed. The numeric DSM has been widely adopted. In a numeric DSM, off-diagonal elements are used to represent not only relationships but also other additional information such as the relative importance or strength of each task dependence [11], [12], and diagonal elements normally represent task durations [13]. It was argued that the binary DSM only provides limited information and the numerical DSM is more appropriate and efficient than the binary DSM [14],[15]. The information flow among design tasks is the research focus of DSM. For example, Krishnan et al.[16] examined the information flow between overlapped tasks and identified two properties: upstream information evolution and downstream iteration sensitivity [17].

	A	B	C	D	E	F
A	A					
B	0.2	B			0.8	
C	0.1		C			0.3
D		0.6	0.2	D	0.4	0.1
E		0.7		0.7	E	
F			0.5			F

	A	C	F	B	D	E
A	A					
C	0.2	C	0.3			
F		0.5	F			
B	0.1			B		0.8
D		0.2	0.1	0.6	D	0.4
E				0.7	0.7	E

Figure 1 A. Design Structure Matrix B. Clustered Design Structure Matrix

2.1 Measuring Cluster Performance "Objective Function"

Numerical Interaction Density (NDd) function is used to evaluate the cluster performance by applying penalty on outer cells. For an n×n matrix, there are n-1 possible clustering results. To select the best solution from all possible clustering results, we need a performance measure to evaluate the clustering performance from each result and determine the final groups. (Lin 2002) [18] Developed a performance measure, Numerical Interaction Density (NDd), to help select the best clustering result. NDd, measuring the numerical interaction strengths outside the block diagonal of the clustered matrix, is formulated as follows: NDd = Ne/OuterCells, applying this on previous illustrated example NDd= (0.1+0.2+0.1)/18 =0.0222

2.2Problem Formulation

2.2.1Solution representation

Solution represented as {1, n clusters} incremented by 1 discrete values representing the id of the cluster for which the element will be included in a vector.

Let $A = \{a_m | a_m \epsilon (1, n\ clusters)\}_{1\times M}$ represents the assignment: $a_m = 1 \rightarrow n$ if element m is assigned to cluster n.

2.2.2 Constraints

Minimum number of elements in cluster are 2

$$\forall A \in Ci \geq 2$$

Number of clusters between 2 and $\frac{n}{2}$

$$\forall\ C \in Ci \geq 2 \leq \frac{n}{2}$$

Where C is the cluster and i $\in$ 1 $\rightarrow$n represents number of clusters, while m is representing number of elements, A is the solution representation and a is the element belongs to which cluster.

3. SOLUTION ALGORITHMS

In this section the algorithm of each optimization technique used had been identified. Those algorithms belong to the evolutionary algorithms (EA) class, which generate solutions for optimizing objective functions inspired by natural evolution.

3.1 Genetic Algorithm (with Integer Constraints)

In a genetic algorithm [19], [20], solutions represented each as string of ones to n reflecting population. Known that's GA initially used to solve binary problems and have its strength in it then customized to solve discrete and continues problems but its strength in binary and discrete problems. GA evolving population in each iteration toward better solution. Starting from random population combined together to generate new better generation. The following figure 2 represents model introduced to cluster designed structured matrix using genetic algorithm optimization with integer constraints applied on solution variables.

```
1: begin
    2: Define Objective function NDd(A).
    3: Input Data as Matrix
    4: Initializing solution that handle assignment constraint and minimizing solution
    which representing cluster assignment for each element
    5: Generate an initial population with random chromosomes based on solution
    representation generated mapped to feasible assignment.
    6: Define parameters based on fine tuning experiments.
    7: while (t < max number of iterations) do
    8:      For all chromosomes in the population, map the jth element in L1 for 1 <
            j < L1. For all m, search all (n, k) that satisfies the constraints
    9:      Evaluate each chromosome according to the objective function NDd(Ai).
    10:     Perform the desired selection and crossover scheme.
    11: end while
    12: Find the current best solution.
13: end
```

Figure 2 Genetic Algorithm

3.2 Binary Particle Swarm Optimization PSO

PSO reflecting swarm behavior which is guided each particle toward best solution in the swarm taking into consideration its behavior and its history based on local best and common global best. PSO originally designed for solving problems with continues space and continues solution values. Which forced (Kennedy, J.; Eberhart, R) [21] the founder of PSO for enhancing the PSO to solve a binary problems. Binary version of PSO transacted with velocity as direction in probability of sigmoid shape (generated velocity belongs to which direction). In binary version of PSO algorithm, particle's velocity defined in continuous space. The update function of particle's velocity is given in below Equation 4

$$v_{ij}^{t+1} = wv_i + \alpha_1 C_1 (pb_i^t) + \alpha_1 C_1 (gb^t - x_i^t) \qquad (4)$$

Where w is an inertia weight, α_1 and α_2 defined as random numbers uniformly distributed between 0 and 1; C1 and C2 are behavior and social parameters, $\mathbf{pb_{ti}}$ local best at iteration t, gb_t global best at iteration t and x_{ti} is position for variable i at iteration t.

Converting continues space into discrete space using special sigmoid function $(\text{sig}\ (\text{vi}(t)))$ in order to calculate the $\mathbf{v}$ using the velocity calculated above.

$$\bar{v}_i^{t+1} = sig\big(v_i(t)\big) = \frac{1}{1 + e^{-v_i(t)}} \qquad (5)$$

Update of particle's position based on:

$$x_i^{t+1} \begin{cases} 1 & if\ r_{ij} < \bar{v}_i^{t+1}, \\ 2 & \text{otherwise.} \end{cases} \qquad (6)$$

Where r_{ij} are uniformly random numbers between 0 and 1. The following figure 2 represents model used to solve cluster allocation problem using Discrete Particle Swarm Optimization.

1: **begin**
2: Define Objective function NDd(A).
3: Input Data as Matrix
4: Initializing solution that handle assignment constraint and minimizing solution which representing cluster assignment for each element
5: Define parameters (number of particles, maximum number of iterations, and P_{mut}).
6: Generate an initial particles with random chromosomes based on solution representation generated mapped to actual position.
7: while (t < max number of iterations) do
8: For all chromosomes in the population, map the jth element in L1 for $1 < j < L1$. For all m, search all (n, k) that satisfies all constraints
9: Evaluate each chromosome according to the objective function NDd(Ai).
10: If a particle's current position is better than its previous best position, update it.
11: Determine the best particle (according to the particle's previous best positions).
12: Update particles' velocities according to Equation 4.
13: Calculate v_i^{t+1} for particle i using Equation 5.
14: Move particles to their new positions according to Equation 6.
15: end while
16: Find the current best solution.
17: **end**

Figure 3 Binary PSO Algorithm

As represented in previous algorithm it is quantified to solve binary solution variables or the solution on our case with two clusters cluster 1 or cluster 2 which fail to represent more than two clusters in Particle Swarm Optimization.

4. NUMERICAL EXPERIMENTS

This section presents and analyzes the numerical experimental for optimizing the original aggregated matrix in [10] to bring minimum NDd and compare it with the original results, then optimizing each interaction function to know its optimal solution and compare it with generated from GA, and PSO algorithms with respect to the minimizing the NDd utility functions.

4.1 Original Aggregated Interactions

Shi-Jie (Gary) Chen Enzhen Huang [10] had converted the three symmetric numerical DSMs from Table.3, 5 and 7 to combine aggregated DSM by assigning a weight to each interaction type. For example, the DSM with information interactions is given a 0.5 weight because the managers and experts consider the information flows are to be more important than the other two interaction types in the supply chain system. The other two interaction types, material and functional interactions, are assigned weights of 0.3 and 0.2 respectively in this example. With the weights assigned, the value of each row i and column j in the aggregated numerical DSM is calculated by Eq. (1):

$$AggregateDSM_{i,j} = InformationDSM_{i,j} * 0.5 + Material_{i,j} * 0.3 + FunctionDSM_{i,j} * 0.2$$

Which resulted in one matrix in Table. 9 with single objective of minimizing its NDd as shown in Table.8 having fitness value NDd equal to 0.0681.

Table 1 Systems elements [10]

1	Plan: Business Strategy
2	Plan: Resource Allocation
3	Plan: Coordination & Communication
4	Source: Delivery Scheduling
5	Source: Supplier Selection
6	Source: Inv Mang
7	Produce: Produ Design
8	Produce: Ind Parts proce
9	Produce: Subsys
10	Produce: WIP/FG
11	Warehouse: Inv Ctrl
12	Warehouse: Procu

13	Warehouse: Carrier Selection
14	Warehouse: Capacity & Operation
15	Deliver: Distribution
16	Deliver: Schedule
17	Order: quotes
18	Order: Processing Order
19	Order: Back orders
20	Order: Invoice
21	Return: receive and verify
22	Return: defective rewark/dispose

Table 2 Function interaction matrix [10]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0.5	0	0.8	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	0	0.8	0	0	0.7	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0.8	0	0.8	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0.6	0	0	1	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	1	0	0.6	0	1	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0.6	0.8	0	0	0	0.3	0	0	0.8	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0.6	0	0	0	0	0.8	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
11	1	0.4	0	0	0	0	0	0	0	0	0	1	0.2	0	0	0	0	0	0	0	0	0
12	0.5	0.2	1	0	0	0	0	0	0	0.2	0.8	0	0	0.1	0	0	0	0	0	0	0	0
13	1	0.2	0	0	0	0	0	0	0	0	0.4	0.5	0	0.5	0	0	0	0	0	0	0	0
14	1	0.6	0	0	0	0	0	0	0	0	0.9	0.2	0	0	0	0	0	0	0	0	0	0
15	1	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
17	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0
19	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	1	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0

Table 3 Function interaction Symmetric matrix [10]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0.3	0	0	0.5	0	0.8	0	0	0	0.5	0.3	0.5	0.5	1	0	0	0	0	0	0	0
2	0.3	0	0.5	0	0.5	0.4	0.3	0.5	0.3	0.3	0.2	0.1	0.1	0.3	0	0.3	0	0	1	0	0	0
3	0	0.5	0	0.3	0	0	0	0	0.4	0	0	0.5	0	0	0	0	0	0	0	0	0	0
4	0	0	0.3	0	0.8	0.9	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0.5	0.5	0	0.8	0	0	0.9	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0.4	0	0.9	0	0	0	0.5	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0
7	0.8	0.3	0	0	0.9	0	0	0.3	0.2	0.4	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0.5	0	0.3	0.1	0.5	0.3	0	0.3	0.5	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0.3	0.4	0	0	0	0.2	0.3	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0.3	0	0	0	0.3	0.4	0.5	0.4	0	0	0.1	0	0.5	0	0	0	0	0	0	0	0
11	0.5	0.2	0	0	0	0	0	0	0	0	0	0.9	0.3	0.5	0	0	0	0	0	0	0	0
12	0.3	0.1	0.5	0	0	0	0	0	0	0.1	0.9	0	0.3	0.2	0	0	0	0	0	0	0	0
13	0.5	0.1	0	0	0	0	0	0	0	0	0.3	0.3	0	0.3	0	0.5	0	0	0	0	0	0

International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 5, Issue 11, November 2016 **ISSN 2319 - 4847**

14	0.5	0.3	0	0	0	0	0	0	0	0.5	0.5	0.2	0.3	0	0	0	0	0	0	0	0	0
15	0.5	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0.3	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	1	0	0	0	0
17	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8	0	0	1	1	0	0
19	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	1	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0

Table 4 Material interaction matrix [10]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0.5	0	0.8	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0.8	0	0	0.7	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0.8	0	0.8	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	1	0	0.6	0	1	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0.6	0.8	0	0	0	0.3	0	0	0.8	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0.6	0	0	0	0	0.8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0.5	0.4	0	0	0	0	0	0	0	0	0	0	0.2	1	0	0	0	0	0	0	0	0
12	0	0.2	1	0	0	0	0	0	0	0.2	0.8	0	0	0.1	0	0	0	0	0	0	0	0
13	0	0.2	0	0	0	0	0	0	0	0	0.4	0.5	0	0	0	0	0	0	0	0	0	0
14	0	0.6	0	0	0	0	0	0	0	0	0.9	0.2	0	0	0	0	0	0	0	0	0	0
15	1	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
17	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
19	0.2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0

Table 5 Material interaction Symmetric matrix [10]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0	1	0	0	0	0	0	0	0
2	0	0	0.2	0	0	0.4	0.3	0.5	0.3	0.3	0.2	0.1	0.1	0.3	0	0	0	0	1	0	0	0
3	0	0.2	0	0.3	0	0	0	0	0.4	0	0	0.5	0	0	0	0	0	0	0	0	0	0
4	0	0	0.3	0	0.8	0.9	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0.8	0	0	0.4	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0.4	0	0.9	0	0	0	0.5	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0.3	0	0	0.4	0	0	0	0.2	0.4	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0.5	0	0.3	0.1	0.5	0	0	0.3	0.5	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0.3	0.4	0	0	0	0.2	0.3	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0.3	0	0	0	0.3	0.4	0.5	0.4	0	0	0.1	0	0	0	0	0	0	0	0	0	0
11	0.3	0.2	0	0	0	0	0	0	0	0	0	0.4	0.3	1	0	0	0	0	0	0	0	0
12	0	0.1	0.5	0	0	0	0	0	0	0.1	0.4	0	0.3	0.2	0	0	0	0	0	0	0	0
13	0	0.1	0	0	0	0	0	0	0	0	0.3	0.3	0	0	0	0.5	0	0	0	0	0	0
14	0	0.3	0	0	0	0	0	0	0	0	1	0.2	0	0	0	0	0	0	0	0	0	0

15	0.5	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0
17	0	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
19	0.1	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0

Table 6 Information interaction matrix [10]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0.6	0	1	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0
3	0.2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0.5	0	0.8	0.9	0	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0.6	0	0	1	0	0	0.7	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	1	0	0.8	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0
7	0.4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0.6	0	0	0	0	0.2	0	0.5	0.2	0	0	0	0	0	0	0	0	0	0	0	0
9	0	1	0.8	0	0	0	1	0.6	0	0.8	0	0	0	0	0	0	0	0	0	0	0	0
10	0	1	0	0	0	0	0.4	0	0	0	0.2	0.6	0	0	0	0	0	0	0	0	0	0
11	0.9	0.4	0	0	0	0	0	0	0	1	0	0	0.2	0	0	0	0	0	0	0	0	0
12	0.7	0.6	0	0	0	0	0	0	0	0.2	0.8	0	0	0.1	0	0	0	0	0	0	0	0
13	0.6	0.2	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0
14	1	0	0	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	0	0
15	1	0.4	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
17	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.2	0	0	0	0	0	0
18	0.4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0

Table 7 Information interaction Symmetric matrix [10]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0.5	0.1	0	0.3	0	0.2	0	0	0	0.5	0.4	0.3	0.5	1	0	0	0	0	0	0.5	0
2	0.5	0	1	0.5	0	0.5	0.5	0.3	0.5	0.5	0.2	0.3	0.1	0.1	0	0	0	1	0	0	0	0
3	0.1	1	0	0.3	0	0	0	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0.5	0.3	0	0.9	0.9	0	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0.3	0	0	0.9	0	0	0.4	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0.5	0	0.9	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0
7	0.2	0.5	0	0	0.4	0	0	0.1	0.5	0.2	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0.3	0	0.1	0.1	0	0.1	0	0.6	0.1	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0.5	0.4	0	0	0	0.5	0.6	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0.5	0	0	0	0.3	0.2	0.1	0.4	0	0.6	0.4	0	0	0	0	0	0	0	0	0	0
11	0.5	0.2	0	0	0	0	0	0	0	0.6	0	0.4	0.1	0.5	0	0	0	0	0	0	0	0
12	0.4	0.3	0	0	0	0	0	0	0	0.4	0.4	0	0.3	0.1	0	0	0	0	0	0	0	0
13	0.3	0.1	0	0	0	0	0	0	0	0	0.1	0.3	0	0	1	0	0	0	0	0	0	0
14	0.5	0.1	0	0	0	0	0	0	0	0	0.5	0.1	0	0	0	0	0	0	0	0	0	0
15	0.6	0.2	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	1	0	0	0	0	0

16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
17	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.1	0	0	0	0	0	0
18	0.2	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	1	1	0	0
19	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
21	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0

Table 8 Aggregated Matrices 3 Clusters NDd=0.0681 [10]

	1	11	12	13	14	4	5	6	15	2	3	7	8	9	10
1	0	0.4	0.4	0.25	0.35	0	0.25	0	0.55	0.3	0.05	0.25	0	0	0
11	0.4	0	0.5	0.2	0.6	0	0	0	0	0.2	0	0	0	0	0.3
12	0.4	0.5	0	0.25	0.1	0	0	0	0	0.2	0.25	0	0	0	0.25
13	0.25	0.2	0.25	0	0.05	0	0	0	0.25	0.1	0	0	0	0	0
14	0.35	0.6	0.1	0.05	0	0	0	0	0	0.2	0	0	0	0	0.1
4	0	0	0	0	0	0	0.85	0.85	0	0.25	0.25	0	0.2	0	0
5	0.25	0	0	0	0	0.85	0	0	0	0.1	0	0.45	0.1	0	0
6	0	0	0	0	0	0.85	0	0	0	0.45	0	0	0.25	0	0.25
15	0.55	0	0	0.25	0	0	0	0	0	0.15	0	0	0	0	0
2	0.3	0.2	0.2	0.1	0.2	0.25	0.1	0.45	0.15	0	0.65	0.4	0.4	0.4	0.4
3	0.05	0	0.25	0	0	0.25	0	0	0	0.65	0	0	0	0.4	0
7	0.25	0	0	0	0	0	0.45	0	0	0.4	0	0	0.1	0.4	0.3
8	0	0	0	0	0	0.2	0.1	0.25	0	0.4	0	0.1	0	0.4	0.3
9	0	0	0	0	0	0	0	0	0	0.4	0.4	0.4	0.4	0	0.4
10	0	0.3	0.25	0	0.1	0	0	0.25	0	0.4	0	0.3	0.3	0.4	0

4.2 Optimizing Aggregated Interactions

An optimization using evolutionary algorithms had been applied and better results had been obtained with NDd= 0.0629 for 3 clusters and an even better results with 2 clusters with NDd= 0.0481

Table 9 Optimized Aggregated Interaction 3 Clusters with NDd=0.0629

	1	11	12	14	2	3	4	5	6	7	8	9	10	13	15
1	0	0.4	0.4	0.35	0.3	0.05	0	0.25	0	0.25	0	0	0	0.25	0.55
11	0.4	0	0.5	0.6	0.2	0	0	0	0	0	0	0	0.3	0.2	0
12	0.4	0.5	0	0.1	0.2	0.25	0	0	0	0	0	0	0.25	0.25	0
14	0.35	0.6	0.1	0	0.2	0	0	0	0	0	0	0	0.1	0.05	0
2	0.3	0.2	0.2	0.2	0	0.65	0.25	0.1	0.45	0.4	0.4	0.4	0.4	0.1	0.15
3	0.05	0	0.25	0	0.65	0	0.25	0	0	0	0	0.4	0	0	0
4	0	0	0	0	0.25	0.25	0	0.85	0.85	0	0.2	0	0	0	0
5	0.25	0	0	0	0.1	0	0.85	0	0	0.45	0.1	0	0	0	0
6	0	0	0	0	0.45	0	0.85	0	0	0	0.25	0	0.25	0	0
7	0.25	0	0	0	0.4	0	0	0.45	0	0	0.1	0.4	0.3	0	0
8	0	0	0	0	0.4	0	0.2	0.1	0.25	0.1	0	0.4	0.3	0	0
9	0	0	0	0	0.4	0.4	0	0	0	0.4	0.4	0	0.4	0	0
10	0	0.3	0.25	0.1	0.4	0	0	0	0.25	0.3	0.3	0.4	0	0	0

Volume 5, Issue 11, November 2016 ISSN 2319 - 4847

13	0.25	0.2	0.25	0.05	0.1	0	0	0	0	0	0	0	0	0	0.25
15	0.55	0	0	0	0.15	0	0	0	0	0	0	0	0	0.25	0

Table 10 Optimized Aggregated Interaction 2 Clusters with NDd=0.0481

	1	11	12	13	14	15	2	3	4	5	6	7	8	9	10
1	0	0.4	0.4	0.25	0.35	0.55	0.3	0.05	0	0.25	0	0.25	0	0	0
11	0.4	0	0.5	0.2	0.6	0	0.2	0	0	0	0	0	0	0	0.3
12	0.4	0.5	0	0.25	0.1	0	0.2	0.25	0	0	0	0	0	0	0.25
13	0.25	0.2	0.25	0	0.05	0.25	0.1	0	0	0	0	0	0	0	0
14	0.35	0.6	0.1	0.05	0	0	0.2	0	0	0	0	0	0	0	0.1
15	0.55	0	0	0.25	0	0	0.15	0	0	0	0	0	0	0	0
2	0.3	0.2	0.2	0.1	0.2	0.15	0	0.65	0.25	0.1	0.45	0.4	0.4	0.4	0.4
3	0.05	0	0.25	0	0	0	0.65	0	0.25	0	0	0	0	0.4	0
4	0	0	0	0	0	0	0.25	0.25	0	0.85	0.85	0	0.2	0	0
5	0.25	0	0	0	0	0	0.1	0	0.85	0	0	0.45	0.1	0	0
6	0	0	0	0	0	0	0.45	0	0.85	0	0	0	0.25	0	0.25
7	0.25	0	0	0	0	0	0.4	0	0	0.45	0	0	0.1	0.4	0.3
8	0	0	0	0	0	0	0.4	0	0.2	0.1	0.25	0.1	0	0.4	0.3
9	0	0	0	0	0	0	0.4	0.4	0	0	0	0.4	0.4	0	0.4
10	0	0.3	0.25	0	0.1	0	0.4	0	0	0	0.25	0.3	0.3	0.4	0

5 CONCLUSION

The Design Structure Matrix acts as a tool to analyze various supply chain elements and subsystem elements and its interaction to gives focus area for the optimization leading to better cluster related subsystem elements in Material, Function and Information minimizing the numerical interaction density (NDd) determining best cluster size and subsystem interdependent cluster the strongly coupled tasks into same group. In summary, effectively managed way in DSM has the previous research in DSM best fit in representing and simplifying large scale supply chain systems. In Supply Chain Network DSM identify the interdependencies within the system. Numerical DSM with quantifiable measures represent the degrees of interactions among system elements in supply chain. Finally clustering technique helps to decompose a complex supply chain system into smaller and manageable components or sub-systems. This paper better ordered and clustered subsystem elements using evolutionary algorithms the best clustered obtained from previous work was three clusters with NDd equal to 0.0681 after optimizing it using Genetic algorithm with three clusters a better NDd result obtained equal to 0.629 even better when optimizing with dynamic number of clusters reach best cluster size to be two clusters with NDd equal to 0.0481 when running Genetic algorithm and Binary PSO same results obtained.

References

[1] ERMİŞ, Murat, Ozgur Koray SAHINGOZ, and Füsun ÜLENGİN. "Mobile agent based supply chain modelling with neural network controlled services." (2004).

[2] Falcone, Marco Aurelio, Heitor Silverio Lopes, and Leandro dos Santos Coelho. "Supply chain optimisation using evolutionary algorithms." International Journal of Computer Applications in Technology 31.3-4 (2008): 158-167.

[3] Zhou, Zhangyu, Siwei Cheng, and Ben Hua. "Supply chain optimization of continuous process industries with sustainability considerations." Computers & Chemical Engineering 24.2 (2000): 1151-1158.

[4] Grossmann, Ignacio E. "Challenges in the new millennium: product discovery and design, enterprise and supply chain optimization, global life cycle assessment." Computers & Chemical Engineering 29.1 (2004): 29-39.

[5] Fu, Michael C. "Optimization for simulation: Theory vs. practice." INFORMS Journal on Computing 14.3 (2002): 192-215.

[6] Terzi, Sergio, and Sergio Cavalieri. "Simulation in the supply chain context: a survey." Computers in industry 53.1 (2004): 3-16.

[7] Steward, Donald V. "The design structure system: A method for managing the design of complex systems." IEEE transactions on Engineering Management 3 (1981): 71-74.

[8] Shi, Qian, and Tomas Blomquist. "A new approach for project scheduling using fuzzy dependency structure matrix." International Journal of Project Management 30.4 (2012): 503-510.
[9] Luh, Ding-Bang, Yao-Tsung Ko, and Chia-Hsiang Ma. "A dynamic planning approach for new product development." Concurrent Engineering 17.1 (2009): 43-59.
[10] Chen, S.-J. G., & Huang, E. (2007). A systematic approach for supply chain improvement using design structure matrix. Journal of Intelligent Manufacturing, 285--299.
[11] S. D. Eppinger, D. E. Whitney, R. P. Smith, et al. Organizing the tasks in complex design projects. Lecture Notes in Computer Science, 492 (1991), 229-252.
[12] R. P. Smith, S. D. Eppinger. Identifying controlling features of engineering design iteration. Management
[13] S. D. Eppinger, D. E. Whitney, R. P. Smith, et al. A model-based method for organizing tasks in product development. Research in Engineering Design, 6:1 (1994), 1-13.
[14] S. J. Chen, L. Lin. Decomposition of interdependent task group for concurrent engineering. Computers & Industrial Engineering, 44:3 (2003), 435-459.
[15] S. J. Chen, E. Huang. A systematic approach for supply chain improvement using design structure matrix. Journal of Intelligent Manufacturing, 18:2 (2007), 285-299.
[16] V. Krishnan, S. D. Eppinger, D. E. Whitney. A model-based framework to overlap product development activities. Management Science, 43:4 (1997), 437-451. Science, 43:3 (1997), 276-293.
[17] Shou, Yongyi, and Chunjiang Song. "An integrated genetic algorithm to optimize engineering information flows through task splitting and partitioning." Information 14.3 (2011): 745-752.
[18] Lin, Li. "A project task coordination model for team organization in concurrent engineering." Concurrent Engineering 10.3 (2002): 187-203.
[19] Deb, Kalyanmoy. "An efficient constraint handling method for genetic algorithms." Computer methods in applied mechanics and engineering 186.2 (2000): 311-338.
[20] Deep, Kusum, et al. "A real coded genetic algorithm for solving integer and mixed integer optimization problems." Applied Mathematics and Computation212.2 (2009): 505-518.
[21] Eberhart, Russ C., and James Kennedy. "A new optimizer using particle swarm theory." Proceedings of the sixth international symposium on micro machine and human science. Vol. 1. 1995.

AUTHOR

AbdoulRahman Mahmoud AlShaar received the B.S. degree in Operation Research and Decision Support System from Cairo University Faculty of Computers and Information 2010. During 2010-2012, he stayed in Decision Support and Future Studies Center, Cairo University to study Optimization techniques and its applications. He now with SAS Statistical Analytics Solution, Inc.

Hisham M. Abdelsalam is a Professor of Operations Research and Decision Support and Director of the Decision Support and Future Studies Center in the Faculty of Computers and Information, Cairo University. He holds a Master of Science and PhD in Mechanical Engineering from Old Dominion University, Norfolk, Virginia, USA.