A Particle Swarm Optimization for Solving Resource Constrained Design Structure Matrix

Hayam G. Mohamed, Mohamed H. Rasmy, Hisham M. E. Abdel Salam
Operations Research and Decision Support Department
FCI, Cairo University
Cairo, Egypt

Abstract
During the past two decades, the importance of research and development (R&D) as an organizational competitive advantage became well recognized worldwide. Managing complex R&D projects requires methodologies and tools different than those used in managing other types of projects. One of these tools is the Design Structure Matrix. This paper presents a particle swarm optimization algorithm for solving resource-constrained research and development projects (R&D) scheduling problems. The algorithm aims to find the optimal sequence of project activities and uses design structure matrix to represent the project under investigation. Results obtained showed an improved performance compared to earlier studies.

Keywords, DSM (Design Structure Matrix), RCPSP (Resource Constrained Project Scheduling Problems, SOF (Shortest Operation First), LOF (Longest Operation First), FCFS (First Come First Served), PSO (Particle Swarm Optimization), MINSLK (Minimum Slack First), PFSP (Permutation Flow Shop Problem).
1. INTRODUCTION

Research and development (R&D) projects are inherently complex; making effective management of the tasks, resources, and teams necessary to bring new products to market is problematic. Frequently, managers of R&D projects are overwhelmed by complicating factors such as stochastic task times, ill-defined specifications, complex interrelationships between tasks, and information dependencies. Moreover, these projects are information-based which means that activity execution and results are based mainly on information exchanged among different coupled activities, and its highly interdependent activities which must converge iteratively to an acceptable solution. Unfortunately, the traditional Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM) fail significantly in research and development projects since they do not tolerate feedback relationships. During the past two decades, several researchers have developed an alternative project management tool called the Design Structure Matrix (DSM) that explicitly takes into account the iterative nature of research development projects [1].

This paper aims to develop a Particle Swarm Optimization (PSO) algorithm to find the optimal sequence (arrangement of activities) within a resource constrained DSM. In this context, the DSM will work as a system analysis tool, which provides a compact and clear representation of a complex system. It captures the interactions/interdependencies/interfaces between system elements. It also works as a project management tool which provides a project representation that allows for feedback and cyclic activity dependencies.

The problem being tackled here is a NP-hard problem. Hence, the enumeration of all feasible combinations is infeasible, and mathematical models will be inadequate for solving it. Heuristics approaches are the most appropriate approaches that can deal with these kinds of problems. In this research, Particle Swarm Optimization Algorithm (PSO) will be used to solve this problem.

Following the introduction section, the rest of this paper is organized as follows. Section 2 provides a brief introduction of Design Structure Matrix, section 3 provides the related work of a resource constrained DSM optimization model. Section 4 presents the proposed solution approach followed by section 5 which provides the results obtain and finally, conclusions and future work are given in section 6.
2. DESIGN STRUCTURE MATRIX

The basic DSM is a simple binary (A cell can hold one of only two values (0, 1), or (an “x” mark, an empty cell)) \( n \)-square matrix (where \( n \) is the number of system elements), with \( m \) non-empty elements (where \( m \) is the number of couplings among different system elements). A typical DSM is shown in Figure 1. Activity names are placed on the left hand side of the matrix as row headings and across the top row as column headings in the same order (order of their execution); a main DSM assumption is that activities are undertaken in the order listed from top to bottom, an off-diagonal mark (x) represents a coupling (an information flow, or a dependency) between two activities. If an activity \( i \) depends on (receives information from) activity \( j \), then the matrix element (row \( i \), column \( j \) ) \( i \ j \) contains an off diagonal mark (x) otherwise the cell is empty.

Marks below the diagonal (sub-diagonal marks) are indicative of feed-forward couplings (i.e. from upstream activities to downstream activities), while those above the diagonal (super-diagonal) represent feedback couplings (i.e. from downstream activities to upstream activities). As they imply iterations, the latter type of couplings should be eliminated if possible or reduced to the maximum extent. If certain feedback couplings cannot be eliminated, the activities can be grouped into iterative sub-cycles (or circuits). For example, in Figure 1 activities (1, 2, 3) and activities (6, 7, 8, 9, 10) are grouped into two iterative sub-cycles. A primary goal in basic DSM analysis is to minimize the number of feedbacks and their scope by restructuring or re-architecting the process [1]. In other words, by re-sequencing the execution of the activities to get the DSM into as lower-triangular form as possible. To achieve this goal, Steward [2,3] proposed a two phase approach: partitioning and tearing. In addition to Steward’s partitioning heuristic, several methods are found in literature: the Path Searching method [4], the Reachability Matrix method [5], the Triangularization Algorithm [6] and the Powers of the Adjacency Matrix Method [7].
3. RELATED WORK

In this section we discuss the related works of our work Particle Swarm Optimization model to a resource constrained design structure matrix. But in literature we cannot find this combination optimization with resource constrained Design Structure Matrix. So, we divided the related work section to two sub section, section 1 optimization on DSM, section 2 DSM with resource constrained, section 3 summarizes past work.

3.1 Optimization on DSM

Rogers [8] at NASA-Langley developed a knowledge-based tool, called the Design Manager's Aid for Intelligent Decomposition with Genetic Algorithm (or DeMAID/GA). The software include functions for minimizing the number of feedback couplings, sequencing the design processes, grouping processes into iterative sub-cycles, and optimizing the sequence of processes within an iterative sub-cycle using a genetic algorithm. DeMAID/GA is characterized by its high functionality in re-ordering the activity to an optimal (or near optimal) solution, decomposing the project into several sub-circuits. Furthermore, a rather important feature of DeMAID/GA is the ability to trace expected changes in the output as a result of a change in an input or more. But it should be noticed that the GA reordering optimization takes place after partitioning, i.e. it optimizes the order of activities within each circuit rather than optimizing the order of activities with respect to the system (DSM) as a whole.
Altus et al. [9] described a method for structuring problem activities with optimal ordering and decomposition into sub-problems. The method further uses Genetic Algorithm and incorporates it into a computer program called AGENDA-A GENetic algorithm for Decomposition of Analyses - with the objective of reducing the extent of feedbacks. Or in other words, minimizing the “total length of feedback” of the system.

Cho [10] introduced an integrated project management framework. The framework consists of three modules. The first module is a DSM-based analysis of the project in which activities are sequenced to have minimum feedbacks from a structural view (partitioning). Finally, Abdelsalam and Bao [11] presented a simulation-based optimization framework that determines the optimal sequence of activities execution within a product development project that minimizes project total iterative time given stochastic activity durations.

### 3.2 DSM with Resource Constrained

Yifeng Zhang, 2008 presented a simulation–based optimization model for solving resource-constrained product development project scheduling problems. The model uses the design structure matrix (DSM) to represent the information exchange among various tasks of the project, modeling based on DSM allows iteration to take place. Adding resource factors to DSM simulation, he not only model the constraints posed by resource requirements, but also explore the effect of allocating different amount of resources on iterations. We compare our work with Yifeng Zhang simulation model but we exceed in optimization model which provides the optimal sequence within Resource Constrained DSM.

Several branch and bound algorithms have been developed to solve scheduling problem with resource constraints [12] [13]. Researchers have formulated efficient bounds that cut down the computation while traversing a branch or backtracking. These algorithms assume acyclic networks, which does not account for iterations.

Many heuristics based approaches can also be found in literature. Most frequently used are priority rules such as first come first serve (FCFS), Shortest operation first(SOF), minimum slack (MINSLK), etc. These rules are applied to different RCPSPs according to the different characteristics of the individual problems. Design of experiments approach is used to study how well each priority rule work for a specific type of project [14]. However, most of these heuristics is difficult. It is also not guaranteed that a single rule will work optimally when used for the whole project span.
In this research, we develop Particle Swarm Optimization model to find the optimal sequence within resource constrained DSM with minimum expected time. Adding to the previous work, the resource-constrained assignment, different type of distribution allowed for each activity duration, and concurrency assumption in achieving the activity.

4. **METHODOLOGY**

In this section we discuss the methodology used to find the optimal sequence within the resource constrained DSM.

4.1 **Particle Swarm Optimization**

PSO algorithms were proposed in the mid-1990s and they are one of the latest evolutionary optimization techniques. Their biological inspiration is based on the metaphor of social interaction and communication in a flock of birds or school of fishes. In these groups, there is a leader who guides the movement of the whole swarm. The movement of every individual is based on the leader and on his own knowledge. Since PSO is population-based and evolutionary in nature, the individuals (i.e. particles) in a PSO algorithm tend to follow the leader of the group, i.e. the one with the best performance. In general, the model that PSO is inspired assumes that the behavior of every particle is a compromise between its individual memory and a collective memory [15].

4.2 **Model characteristics**

The model characteristics are as follow:

- Time of the activity is stochastic and follows any distribution according to user preferences like (e.g. Triangular, uniform).
- Concurrency is allowed in performing the activities.
- Limited amount of resources available.
- Several types of resources are allowed.
- Total amount of resource available is more than any single activity requirements; at any time, at least one activity can be executed.
- SOF criterion is used to resolve resource conflict.
- Learning curve (LC) is a measure of the activity when it repeats; when activity does work for more than one time the duration of that activity is represented by learning curve.
- Logical constrained are allowed in output sequence [11].

4.3 **Algorithmic Steps**

The idea of solving this problem came up to us from another problem called Permutation Flow Shop Problem (PFSP) [16]. The complete computational procedure of the PSO algorithm for the problem can be summarized as follows:
**Inputs:** input variables of the project.

**Outputs:** the optimal sequence within the resource constrained DSM with minimum expected time.

**Step 1: Initialization**
- Set $t=0$, $NP=$ twice the number of dimensions.
- Generate $NP$ particles randomly as explained before, $\{X_i^0,i=1,2,\ldots,NP\}$ where $X_i^0=\{x_{i1}^0,x_{i2}^0,\ldots,x_{in}^0\}$.
- Generate the initial velocity for each particle randomly, $\{V_i^0,i=1,2,\ldots,NP\}$ where $V_i^0=\{v_{i1}^0,v_{i2}^0,\ldots,v_{in}^0\}$.
- Apply the SPV rule to find the permutation $\pi_i^0=\{\pi_{i1}^0,\pi_{i2}^0,\ldots,\pi_{in}^0\}$ of particle $X_i^0$ for $i=1,2,\ldots,NP$. We must take into consideration the logical constrained (i.e. the first activity and the last activity in the sequence are known).
- Evaluate each particle in the swarm using the objective function, $f_i^0$ for $i=1,2,\ldots,NP$. In evaluation we use our simulation model [17].
- For each particle in the swarm, set $P_i^0=X_i^0$, where $P_i^0=\{p_{i1}^0=x_{i1}^0,p_{i2}^0=x_{i2}^0,\ldots,p_{in}^0=x_{in}^0\}$ together with its best fitness value $f_i^{pb}$ for $i=1,2,\ldots,NP$.
- Find the best fitness value among the whole swarm such that $f_i^t=\min\{f_i^0\}$ for $i=1,2,\ldots,NP$ with its corresponding positions $X_i^0$, set global best to $G_0^t=X_i^0$ such that $G_0^t=\{g_1^t=x_{i1},g_2^t=x_{i2},\ldots,g_n^t=x_{in}\}$ with its fitness value $f_{gb}^t=f_i^t$.

**Step 2: Update Iteration Counter**
- $t=t+1$

**Step 3: Update Inertia Weight**
- $W^t=W^{t-1} \ast \beta$ where $\beta$ is the decrement factor.

**Step 4: update velocity using equation (1)**
$$v_{id} = W^t \ast v_{id} \ast c_1 \ast \text{rand1} \ast (p_{best}^t - x_{id}) + c_2 \ast \text{rand2} \ast (g_{best}^t - x_{id})$$
(1)
Where $c_1$ and $c_2$ are acceleration coefficients and rand1 and rand2 are uniform random numbers between (0,1).

**Step 5: Update Position Using Equation (2)**
$$x_{id} = x_{id} + v_{id}$$
(2)

**Step 6: Find Permutation**
- Apply the SPV rule to find the permutation $\pi_i^t=\{\pi_{i1}^t,\pi_{i2}^t,\ldots,\pi_{in}^t\}$ of particle $X_i^t$ for $i=1,2,\ldots,NP$.

**Step 7: Update the Personal Best**
- Each particle is evaluated by using permutation to see if the personal best will improve, that is if $f_i^t < f_i^{pb}$ for $i=1,2,\ldots,NP$ then personal best is updated as $P_i^t=X_i^t$ and $f_i^{pb}=f_i^t$.

**Step 8: Update the Global Best**
- Find the minimum value in the personal best that is $f_i^t=\min\{f_i^{pb}\}$, $i=1,2,\ldots,NP$ if $f_i^t < f_{gb}^t$, then the global best is updated as $G_i^t=X_i^t$ and $f_{gb}^t=f_i^t$. 

7
**Step 9: Stopping Criterion**

- If the number of iteration exceeds the maximum number of iteration, or maximum CPU time then stop; otherwise go to step 2.

**5. RESULTS**

We present the results obtained using our optimization model in solving numerical example, this example contain a set of activities with stochastic duration follow a triangular distribution, DSM defines the relation feedback and feed–forward between activities, Rework probability matrix, Rework Impact Matrix, resource availability, resource requested for each activities as shown in appendix 1, number of optimization model runs = 100, number of simulation runs = 100, We apply our optimization model to several cases and find results.

**Base Case**, using the example in appendix 1, we find this output as shown in Figure 2 and the optimal sequence is "1 9 3 12 8 2 13 6 10 5 11 7 4 14". The expected total time of the project = 174 days.

![Figure 2](image)

**Figure 2**: The relation between the iteration number and the pbest, gbest solution in each iteration.
5.1 Model validation

In this section we present different scenario to prove that the results of our model are logic.

**Case 1**, If we increase the total amount of available resources, type 1 of resource =50, type 2 of resource=50, Using the example data given in appendix 1, we find this output as shown in Figure 3 and find the optimal sequence is "1 8 10 3 13 2 6 11 12 7 9 5 4 14", the expected total time of the project= 147 days, We note from results that the Expected total time of the project will decrease, which proves the inverse relationship between the total amount of resources and the total time of the project.

![Figure 3: The relation between the iteration number and the pbest, gbest solution in each iteration.](image)

**Case 2**, if we decrease the total amount of available resources type 1 = 5 units, type 2 =6 units, using the example data given in appendix 1, we find this output as shown in Figure 4 and the optimal sequence is"1 8 3 4 13 7 6 10 2 9 11 5 12 14". The expected total time of the project= 194 days. We note from results that the Expected total time
of the project will increase, which proves the inverse relationship between the total amount of resources and the total time of the project.

![Graph](image1.png)

**Figure 4: The relation between the iteration number and the pbest, gbest solution in each iteration.**

**Case 3,** If we increase one type of resources and remain the second to be constant, **Case 3a,** Type one increase to be = 50 units, Type two remaining constant=12 units, Using the example data given in appendix 1, we find this output. The optimal sequence is "1 7 3 8 2 4 6 10 11 13 5 9 12 14". The expected total time of the project= 150 days.

**Case 3b,** Type one =10 units remaining constant, Type two increase to be = 50 units, Using the example data given in appendix 1, we find this output. The optimal sequence is "1 8 9 10 6 2 4 3 7 13 5 12 11 14". The expected total time of the project= 160 days. The Expected total time has decreased so much from the base case since most of activity depends on the second type of resources which are not constrained.
Case 4, in our model we have the flexibility to make each activity follow different type of distribution; we have the following distributions to deal with Gamma, uniform, triangular, Weibull, Normal, Log-Normal, and Poisson. Let's assume the activities in appendix 1 all follow uniform distribution with different min, max values. We find this output as shown in Figure 5 and the optimal sequence is "1 2 4 8 7 3 6 12 11 13 5 10 9 14". The expected total time of the project= 200

![Figure 5: the relation between the iteration number and the best, gbest solution in each iteration.](image-url)
6. CONCLUSION AND FUTURE WORK

In this paper, we have presented a Particle Swarm optimization model for solving RCPSPs. We use DSM as our primary data repository and representation tool. It provides a clear visualization, eases modifications to RCPSPs, and allows us to use many mathematical analytic methods to analyze the problem. More importantly, rework, which commonly occurs in process development projects, can now be easily modeled through the use of rework probabilities and rework impacts using DSM. In our optimization model, we use the previous discrete event simulation model to evaluate the fitness function, we generate each sequence and send to simulation model to find the corresponding fitness function and return again to optimization model, we repeat this process for all sequence until we find the optimal sequence with minimum time. Simulation–based approach allows us to generate according to rework probabilities and rework impacts stochastically, instead of a single value of total project durations. We examine a distribution of total project durations as result of many discrete event simulation runs. We can see clearly how rework evolves and contributes to the total project durations. In the future, we will try to use different heuristics instead of SOF like LOF, FCFS and MINSLK to resolve the resource conflict, find the optimal sequence within the resource constrained DSM, find the minimum time using RCPSPs simulation model, and finally find the maximum utilization of the available resources.
References

Table 1: Activity Data for Model Input

<table>
<thead>
<tr>
<th>ID#</th>
<th>Activities</th>
<th>BCV</th>
<th>MLV</th>
<th>WCV</th>
</tr>
</thead>
<tbody>
<tr>
<td>A4</td>
<td>Prepare UAV Preliminary DR&amp;O</td>
<td>1.9</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>A511</td>
<td>Create UAV Preliminary Design</td>
<td>4.75</td>
<td>5</td>
<td>8.75</td>
</tr>
<tr>
<td>A512</td>
<td>Prepare &amp; Distribute Surfaced Models &amp; Int. Arngmt. Drawings</td>
<td>2.66</td>
<td>2.8</td>
<td>4.2</td>
</tr>
<tr>
<td>A531</td>
<td>Perform Aerodynamics Analyses &amp; Evaluation</td>
<td>9</td>
<td>10</td>
<td>12.5</td>
</tr>
<tr>
<td>A521</td>
<td>Create Initial Structural Geometry</td>
<td>14.3</td>
<td>15</td>
<td>26.3</td>
</tr>
<tr>
<td>A522</td>
<td>Prepare Structural Geometry &amp; Notes for FEM</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>A531</td>
<td>Develop Structural Design Conditions</td>
<td>7.2</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>A532</td>
<td>Perform Weights &amp; Inertias Analyses</td>
<td>4.75</td>
<td>5</td>
<td>8.75</td>
</tr>
<tr>
<td>A533</td>
<td>Perform S&amp;C Analyses &amp; Evaluation</td>
<td>18</td>
<td>20</td>
<td>22</td>
</tr>
<tr>
<td>A5341</td>
<td>Develop Balanced Free body Diagrams &amp; External Applied Loads</td>
<td>9.5</td>
<td>10</td>
<td>17.5</td>
</tr>
<tr>
<td>A5343</td>
<td>Establish Internal Load Distributions</td>
<td>14.3</td>
<td>15</td>
<td>26.3</td>
</tr>
<tr>
<td>A5344</td>
<td>Evaluate Structural Strength, Stiffness, &amp; Life Analyses</td>
<td>13.5</td>
<td>15</td>
<td>18.8</td>
</tr>
<tr>
<td>A54</td>
<td>Preliminary Manufacturing Planning &amp; Analyses</td>
<td>30</td>
<td>32.5</td>
<td>36</td>
</tr>
<tr>
<td>A6</td>
<td>Prepare UAV Proposal</td>
<td>4.5</td>
<td>5</td>
<td>6.25</td>
</tr>
</tbody>
</table>

Figure 1: Rework Probabilities Matrix
**Figure 2: Rework Impacts Matrix**

Number of types of resources=2, Resource Availability (RA) Type 1=10, Type 2=12

Resource requested for each activity (Req)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>A4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A511</td>
<td>2</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A512</td>
<td>3</td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A521</td>
<td>4</td>
<td>0.4</td>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A522</td>
<td>5</td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td></td>
<td>0.3</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>A531</td>
<td>6</td>
<td>0.1</td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>A532</td>
<td>7</td>
<td>0.5</td>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.5</td>
</tr>
<tr>
<td>A533</td>
<td>8</td>
<td></td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5341</td>
<td>9</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5342</td>
<td>10</td>
<td></td>
<td>0.1</td>
<td>0.5</td>
<td>0.4</td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>A5343</td>
<td>11</td>
<td></td>
<td>0.5</td>
<td>0.5</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5344</td>
<td>12</td>
<td>0.5</td>
<td></td>
<td>0.3</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A54</td>
<td>13</td>
<td>0.9</td>
<td></td>
<td></td>
<td>0.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.3</td>
</tr>
<tr>
<td>A6</td>
<td>14</td>
<td>0.5</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
</tbody>
</table>