MPM Job Shop Scheduling Problem: a bi-objective approach

Dimitrios Tselios
School of Computer Science and Informatics
University College of Dublin
Belfield, Dublin 4, Ireland
Email:Dimitrios.Tselios@ucdconnect.ie

Ilias K. Savvas
Dept. of Computer Science and Engineering
TEI of Thessaly
Larissa, Greece
Email: savvas@teilar.gr

M-Tahar Kechadi
School of Computer Science and Informatics
University College of Dublin
Belfield, Dublin 4, Ireland
Email: tahar.kechadi@ucd.ie

***Abstract*—This paper presents a Recurrent Neural Network approach for the multipurpose machines Job Shop Scheduling Problem. This case of JSSP can be utilized for the modelling of project portfolio management besides the well known adoption in factory environment. Therefore, each project oriented organization develops a set of projects and it has to schedule them as a whole. In this work, we extended a bi-objective system model based on the JSSP modelling and formulated it as a combination of two recurrent neural networks. In addition, we designed an example within its neural networks that are focused on the Makespan and the Total Weighted Tardiness objectives. Moreover, we present the findings of our approach using a set of well known benchmark instances and the discussion about them and the singularity that arises.**

Keywords–Recurrent Neural Network; Multipurpose machines; Job Scheduling Problem; Bi-objective; Singularity

I. Introduction

Job Shop Scheduling Problem (JSSP) has been adopted by many researchers as base for their models in order to solve a variety of problems usually coming from manufacturing. Moreover, recent research efforts [26], [27] utilized the JSSP formulation in order to solve project scheduling problems. The original project management focuses on individual projects [2], [25], [20], [3], while on the other hand, the modern project oriented organisations have to operate in multi-project environments [9]. This is a common attribute to the organizations services performing a set of projects concurrently. Hence, the simple project's management approaches are not suitable for the contemporary project management requirements.

Job Shop Scheduling Problem has been investigated by many researchers during the last decades in order to confront analogous problems successfully [15]. Some of these endeavors [13] utilized a quite interesting and promising tool; the recurrent neural network.

In this paper, we extend a recurrent neural network based model [26] for a case of job shop scheduling problem, the multi-purpose machines with identical speeds problem (MPM) that models adequately to the project portfolio. This bi-objective model, beside the Makespan objective function, is focused on the Total Weighted Tardiness as proposed by a recent work [27]. In addition, we test the resulted RNNs using a set of known benchmark instances, and finally proposing a scalar objective as a linear combination of the selected objectives.

The rest of the paper is organized as follows. The related research work is presented briefly in Section II. In Section III we describe succinctly the model, the structure of the RNNs, and two algorithms which improve their evolution. In Section V we provide the experimental results of the neural networks using a set of benchmark instances, and finally, in Section VI we conclude and provide few future research directions.

II. Related Work

During the last three decades many phased models have been designed and applied by the research community and the project management professionals [19], [17]. In addition, some research projects [5], [29] were focused on the formulation of processes on the design of processes for project management in special industries, for instance IT projects. More specifically, there is an extensive list of models [10] for the case of software projects.

The common attribute of those models is that they propose a process framework within a fixed identical number of activities or phases for every project. In other words, according to them, each organization divides its projects into an identical number of discrete phases. This is one of the key characteristics of the Job Shop Scheduling Problem [4].

Sound research has been conducted during the last decades focusing on single objective models of Job Shop Scheduling Problem (JSSP) and the Resource Constrained Scheduling Problem (RCPSP). However, recent research works [12], [23], [30], [21] have asserted that the single objective modelling is insufficient and many problems have to be resolved utilizing the multi-objective approach. The increase of computer performance, the effective application of the metaheuristics on single-objective problems, and the significance of multi-objective approach in many scientific disciplines are three of the factors which [12] have contributed to the adoption of the multi-objective models.

In addition, the above works contend that the multi-objectives models are more effective than single-objective ones in any way. Some of them are based on an approach called Pareto front [21], [12]. Briefly, this approach utilizes a pool of objectives and in each step chooses some from a set that represent the optimum or a near optimum solution at this step. According to some researchers [12], these objectives are independent and non redundant.

On the other hand, other researchers [23], [30] proposed an alternative approach; they constructed a new objective that combines, two or more of the following objectives; makespan, mean tardiness, mean flow time, and mean machine idle time. The final objective function is usually a linear combination of these objectives with arbitrary weights or coefficients that are usually estimated by experimental results [30]. Although this method is quite simple and criticized [31], it has been extensively used.

Smith [22] presented a review work about the adoption of neural networks as a solving tool for the NP-hard combinatorial problems. According to him, most of the past research on the application of Neural Networks on scheduling has been conducted on job-shop scheduling.

One of the most interesting approaches to NNs used so far is Lagrangian Relaxation of the constraints combined with Recurrent Neural Networks (RNNs) [13] usually aiming at one objective function. The key idea of this approach is that the energy function is composed of several terms. However, the main shortcoming of this method is the existence of many local minima.

RNNs have been adopted by many researchers in order to confront combinatorial optimization problems [13]. This adoption described comprehensively in a seminal review work [7]. The concept of this combination is to start from a feasible solution of the problem and its aim is to find better feasible solutions reducing the value of the energy function. Lastly, the method for the solving of a combinatorial problem using a RNN described by Dreyfus [7], has been utilized by recent works [26], [27].

One crucial choice of a neural network is its activation function. During the last decades a vast number of researchers proposed several activation functions for their neural network [8]. The most common function is the Step function [24]. Another useful function that simplifies the equations of motion's calculations is the quadratic function [26].

According to PMI's [19] definition, the successful project management has to accomplish two goals; to deliver the project's deliverables in time and within the budget. Usually, the projects have predefined deadlines that are ensured by imposed severe violation penalties. The measure of those penalties is the Total Weighted Tardiness, one of the choices for the objective functions of the model [27], [14], [31].

Previous research efforts showed that the RNN [26], [27] can provide very promising solutions compared to the solutions devised by the typical heuristic methods. These works provide a model for bi-objective special case of the job scheduling that fits well for project portfolio scheduling.

Very few researchers have designed and tested sets of benchmarks that are suitable for the specific MPMS problem. One of them is B. Jurisch [1] who constructed a set of benchmarks via the modification of the well known benchmarks created by Lawrence [16]. Moreover, this work has been well accepted by the community and exploited in subsequent works [11]. Currently, a lot of research efforts are based on these benchmarks [28], [18]. An interesting set has been proposed as an extension of the Lawrence [16] and Jurisch [1] benchmarks to cover the objective of the Tardiness [6].

III. System Model

The adopted system model [26], [27] for the project portfolio scheduling problem is defined by the following premises [15], [2]:

According to the classification $\alpha|\beta|\gamma$ the simplified version of our problem can be expressed as follows [26]:

$\alpha_1 = PMPM, \alpha_2 = k|\beta_2 = chains, \beta_5 = d_i|\gamma = 0.5 \times C_{max} + 0.5 \times \sum w_i T_i$

Briefly, each schedule S is a vector that consists of the start times of the activities (operations) of all projects (jobs). So, according to the definitions described in previous work [26], [27] the constrained model can be expressed as follows:

$$\min(0.5 \times F_{ms}(S) + 0.5 \times F_{wt}(S)) \Rightarrow$$
$$\min(0.5 \times C_{max} + 0.5 \times \sum_{i=1}^{N} w_i T_i)$$

Subject to

$S_{ix} - S_{ix+1} + \phi_{ix} \leq 0, i \leq N, x \leq X - 1$

$S_{ix} \geq 0, i \leq N, 1 \leq x \leq X$

$\delta^m_{ix,jy}(S^m_{ix} - S^m_{jy} + \phi_{ix}) \leq 0, \forall r_m \in r_{ix} \cap r_{jy}, i \neq j$

$(1 - \delta^m_{ix,jy})(S^m_{jy} - S^m_{ix} + \phi_{jy}) \leq 0, \forall r_m \in r_{ix} \cap r_{jy}, i \neq j$

$S^m_{ix} = S_{ix}, \forall m \in R$

where $\delta^m_{ix,jy} = 1$ if $S^m_{ix} \leq S^m_{jy}$ otherwise $\delta^m_{ix,jy} = 0$. Note that the activities a_{ix} and a_{jy} can be assigned to the same resource r_m. The explanation of every term used in this modelling is presented in table I.

The amended model [26], [27] utilizes separate RNNs for every objective function. These neural networks don't include hidden layers and consist of only one layer of neurons with zero input. The number of the neurons for each RNN is calculated easily by the product $N \times X$ where N is the number of projects and X the number of activities for every project.

The energy functions and the equations of motion for both RNNs, according to the above formulation, defined by the equations (1), (2), (3), (4) are as follows:

$$E_{ms}(S) = F_{ms}(S) + K_{ms} L_{ms}(S) \tag{1}$$

$$E_{wt}(S) = F_{wt}(S) + K_{wt} L_{wt}(S) \tag{2}$$

$$\frac{dS_{ij}}{dt} = -\mu_{ms} \frac{\partial E_{ms}(S)}{\partial S_{ij}} \tag{3}$$

$$\frac{dS'_{ij}}{dt} = -\mu_{wt} \frac{\partial E_{wt}(S)}{\partial S_{ij}} \tag{4}$$

The above functions (1), (2) more analytically are equal to (5), (6),

$$E_{ms}(S) = \max_{i=1}^{N} C_i + K_{ms} \sum_{i=1}^{\xi} \rho_{mk}[c_i(S)] \tag{5}$$

$$E_{wt}(S) = \sum_{i=1}^{N} w_i T_i + K_{wt} \sum_{i=1}^{\xi} \rho_{wt}[c_i(S)] \tag{6}$$

The next step of this modelling is the finding of the first feasible schedule. We used a greedy algorithm to find a good initial schedule. This technique assigns to each activity a resource without violating the constraints. The criterion for the choice is that the shortest first unassigned activity from all projects (jobs) is

Table I
MODELLING PARAMETERS

Symbol	*Definition*
N	Number of projects
i	Index number of project
$P_1, P_2, \ldots, P_N$	The projects
$a_{i1}, a_{i2}, \ldots, a_{iX}$	Activities of project P_i
X	Total number of the activities
x	Index number of a activity
d_i	Deadline time of project P_i
w_i	Penalty cost (weight) of project P_i
a_{ix}	x activity of project P_i
ϕ_{ix}	Load of activity a_{ix}
r_{ix}	Resources that can perform the activity a_{ix}
S_{ix}	Start time of activity a_{ix} for Makespan
S'_{ix}	Start time of activity a_{ix} for Weighted Tardiness
$S_{ix}(t)$	Start time of activity a_{ix} at time t
S^m_{ix}	Start time of activity a_{ix} assigned to resource r_m
C_{ix}	Finish (complete) time of activity a_{ix}
$R = \{r_1, r_2, \ldots, r_M\}$	Set of the resources
M	Number of the resources
m	Index number of a resource
C_i	Scheduling time of project P_i
C_{max}	Scheduling time of the Project Portfolio
F_{ms}, F_{wt}	Objective functions
T_i	Tardiness of project P_i
T	Percent of tardy jobs
D	Due date range
ξ	Total number of the constraints
K_{ms}, K_{wt}	Penalty factors
E_{ms}, E_{wt}	Energy functions
μ_{ms}, μ_{wt}	Learning parameter of the RNNs
U	Uniform distribution
E	Mean function
k	Coefficient of the Uniform distribution
L_1, L_2	Violation's penalties of the constraints
ρ_1, ρ_2	Activation functions of the L_1, L_2

assigned to the first available resource that can execute it. The technique is described by the algorithm which has been proposed in previous relevant work [26], [27]. The key point in this algorithm is the use of the criterion that can be altered with regard to different heuristic rules. According to the first results from the algorithm's running, it gives a feasible initial schedule which is 50% closer to the optimum than a randomly created schedule (without using heuristics) is.

The first running of the RNN evolution showed that the convergence point of every RNN and for every benchmark instance was very far away form the already known minimum or its lower bound. Actually, this point was very close to the processing time (load) of the activity (phase) that has the maximum processing time. Obviously, that RNN's outcome leads to a schedule that it is neither effective nor feasible one.

So, in order to amend this problem, we implemented two adjustments applied to the schedule derived by the RNN in every loop of its evolution. The first adjustment confronts the violation of the conjunctive constraints. Those constraints during the RNN's loop execution are frequently violated. That means that sometimes two consecutive activities (phases) of a job (project) are overlapped timely. The simple technique that is utilized is described by the Algorithm 1.

Algorithm 1 Conjunctive Constraints Adjustment

```
1: for all $P_i \in PP$ do
2:   for all $j \leq X - 1$ do
3:     if $Ph_{i,j}.startime + Ph_{i,j}.load > Ph_{i,j+1}.startime$ then
4:       $Ph_{i,j+1}.startime \leftarrow Ph_{i,j}.startime + Ph_{i,j}.load$
5:     end if
6:   end for
7: end for
```

The second adjustment is aiming at the violation of the disjunctive constraints i.e. the overlapping of two activities belonging to different jobs that are executed at the same time from the same machine (resource). Although the initial schedule algorithm [26] delivers schedules that don't have this type of violation, the RNN's evolution derives after some loops non-feasible schedules. The proposed technique is similar to Algorithm 1 and is presented by Algorithm 2.

Actually, the two techniques tried to keep the schedule derived in every loop feasible. However, there is a contradiction between them. More specifically, the execution of the Algorithm 2 affects the outcome of the Algorithm 1 and vice versa. The consequence is that the overall schedule is not always feasible but the measure of the non-feasibility is already measured by the corresponding energy function.

Algorithm 2 Disjunctive Constraints Adjustment

```
1: for all $Ph_{i,x}, Ph_{j,y}$ involved in a Disjunctive Constraint do
2:   if $Ph_{i,x}.assigned = Ph_{j,y}.assigned$ then
3:     if $Ph_{i,x}.startime + Ph_{i,x}.load > Ph_{j,y}.startime$ then
4:       $Ph_{j,y}.startime \leftarrow Ph_{i,x}.startime + Ph_{i,x}.load$
5:     else
6:       if $Ph_{j,y}.startime + Ph_{j,y}.load > Ph_{i,x}.startime$ then
7:         $Ph_{i,x}.startime \leftarrow Ph_{j,y}.startime + Ph_{j,y}.load$
8:       end if
9:     end if
10:  end if
11: end for
```

Table II
PROJECTS' DEADLINES AND THEIR PENALTY COSTS

Project	*Deadline Time*	*Penalty Cost per time unit*
P_1	5	1
P_2	7	2
P_3	6	3

IV. EXAMPLE

In order to explain clearly the model adopted above we explain its steps through an example. The presented example is more complicated than the example described in previous work [27]. We assume here that an organization manages a Project Portfolio that has three projects; P_1, P_2, P_3. Their predefined deadline times and the corresponding penalty costs are presented in table II.

The organization has a management scheme that divides its projects into three activities i.e. $X = 3$. Moreover, the organization has three resources allocated to this project portfolio; $R = \{r_1, r_2, r_3\}$. Because each activity is characterised by two parameters;

$a_{ix} : \{\phi_{ix}, r_{ix}\}$, the activities of the two projects are characterised as follows:

$a_{11} : \{2.0, \{r_2\}\}$
$a_{12} : \{4.0, \{r_2, r_3\}\}$
$a_{13} : \{1.0, \{r_1\}\}$
$a_{21} : \{3.0, \{r_1\}\}$
$a_{22} : \{2.0, \{r_3\}\}$
$a_{23} : \{2.0, \{r_1, r_2\}\}$
$a_{31} : \{2.0, \{r_1, r_2\}\}$
$a_{32} : \{2.0, \{r_1, r_3\}\}$
$a_{33} : \{1.0, \{r_3\}\}$

The corresponding conjunctive and disjunctive constraints are presented in the diagram shown in Figure 1. The blue lines with an arrow at the one end represent the conjunctive constraints between consecutive activities of a project (job). The red double arrow lines represent the disjunctive constraints between activities belonging to different projects (jobs).

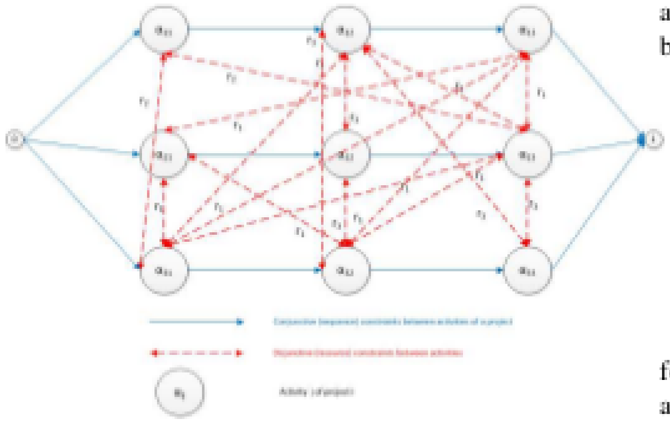
Conjunctive (sequence) constraints between activities of a project
Disjunctive (resource) constraints between activities

Figure 1. The disjunctive diagram of the example

The constrained form of the objective function is as follows.

$$\min(0.5 \times C_{max} + 0.5 \times (1 * T_1 + 2 * T_2 + 3 * T_3))$$

The first of the conjunctive constraints is as follows and the rest of them are similar [26].

$$S_{11} - S_{12} + \phi_{11} \leq 0$$

Similarly, the first pair of the disjunctive constraints is as follows and the rest of them have the analogous form [27].

$$\delta_{11,23}(S_{11} - S_{23} + \phi_{11}) \leq 0$$
$$(1 - \delta_{11,23})(S_{23} - S_{11} + \phi_{23}) \leq 0$$

By converting the constrained problem into an unconstrained one we have the following equations using the transformation proposed in a recent work [13], the first of the conjunctive constraint is transformed as follows.

$$c_1 = S_{11} - S_{12} + \phi_{11}$$

Similarly, the first pair of the disjunctive constraints are transformed as follows.

$$c_7 = \delta_{11,23}(S_{11} - S_{23} + \phi_{11})$$
$$c_8 = (1 - \delta_{11,23})(S_{23} - S_{11} + \phi_{23})$$

Consequently, the two energy functions of the example, using the general forms (5), (6), are described by the equations (7), (8):

$$E_{ms}(S) = C_{max} + K_{ms} \sum_{i=1}^{40} \rho_1[c_i(S)] \tag{7}$$

$$E_{wt}(S) = \sum_{i=1}^{3} w_i T_i + K_{wt} \sum_{i=1}^{40} \rho_2[c_i(S)] \tag{8}$$

The next step calculates the equation of motion for each start time S_{ij} and for each energy function according to the formulas written in section 3. The equation of motion of the first neuron for the first energy function E_{ms} is as follows (the rest have similar form):

$$\frac{dS_{11}}{dt} = -\mu_{ms} \frac{\partial E_{ms}(S)}{\partial S_{11}}$$

Let us calculate the exact form of the first equation of motion:

$$\frac{dS_{11}}{dt} = -\mu_{ms} \frac{\partial E_{ms}(S)}{\partial S_{11}} = -\mu_{ms} \frac{\partial C_{max}}{\partial S_{11}}$$
$$-\mu_{ms} K_{ms} \frac{\partial(\rho_1(c_1) + \rho_1(c_7) + \rho_1(c_8) + \rho_1(c_{23}) + \rho_1(c_{24}))}{\partial S_{11}}$$

We choose the activation function to be of quadratic form. The following gives the details of the above equation of motion:

$$\frac{dS_{11}}{dt} = -\mu_{ms}\frac{\partial C_{max}}{\partial S_{11}} -$$
$$\mu_{ms}K_{ms}[\frac{\partial(1/2(S_{11}-S_{12}+\phi_{11})^2)}{\partial S_{11}} +$$
$$+\frac{\partial(1/2(\delta_{11,23}(S_{11}-S_{23}+\phi_{11}))^2)}{\partial S_{11}}$$
$$+\frac{\partial(1/2((1-\delta_{11,23})(S_{23}-S_{11}+\phi_{23}))^2)}{\partial S_{11}} +$$
$$+\frac{\partial(1/2(\delta_{11,31}(S_{11}-S_{31}+\phi_{11}))^2)}{\partial S_{11}}$$
$$+\frac{\partial(1/2((1-\delta_{11,31})(S_{31}-S_{11}+\phi_{31}))^2)}{\partial S_{11}}] =$$
$$= -\mu_{ms}\frac{\partial C_{max}}{\partial S_{11}} - \mu_{ms}K_{ms}[(S_{11}-S_{12}+\phi_{11}) +$$
$$+\delta^2_{11,23}(S_{11}-S_{23}+\phi_{11}) + (1-2\delta_{11,23}+\delta^2_{11,23})$$
$$\times(S_{23}-S_{11}+\phi_{23}) + \delta^2_{11,31}(S_{11}-S_{31}+\phi_{11})$$
$$+(1-2\delta_{11,31}+\delta^2_{11,31}) \times (S_{31}-S_{11}+\phi_{31})]$$

Finally, the equation of motion is:

$$\frac{dS_{11}}{dt} = -\mu_{ms}\frac{\partial C_{max}}{\partial S_{11}}$$
$$-\mu_{ms}K_{ms}[2(\delta_{11,23}+\delta_{11,31})S_{11} - S_{12}$$
$$+(1-2\delta_{11,23})S_{23} + (1+\delta^2_{11,23}+\delta^2_{11,31})\phi_{11} +$$
$$(1-\delta^2_{11,23})\phi_{23} + (1-2\delta_{11,31})S_{31}$$
$$+(1-\delta^2_{11,31})\phi_{31}]$$

and the first neuron of the RNN is determined by the following equation:

$$S_{11}(t+1) = S_{11}(t) + \frac{dS_{11}}{dt} = S_{11}(t) -$$
$$\mu_{ms}K_{ms}[2(\delta_{11,23}+\delta_{11,31})S_{11}(t) - S_{12}(t)$$
$$+(1-2\delta_{11,23})S_{23}(t) + (1+\delta^2_{11,23}+\delta^2_{11,31})\phi_{11}$$
$$+(1-\delta^2_{11,23})\phi_{23} + (1-2\delta_{11,31})S_{31}(t)$$
$$+(1-\delta^2_{11,31})\phi_{31}]$$

Moreover, according to the neural network formulation a neuron of the network is presented in the Figure 2. More specifically, the starting time of each activity, for instance S_{11}, is represented by a neuron and its value is changed at time t according to the values of the rest involved neurons at time $t-1$. The rest of the equations of motion can be calculated in a similar way.

Actually, the rest of the equations of motion are analogous but one. The only one neuron which is different than the other neurons is that one from S_{13}, S_{23}, and S_{33} which represents the activity with completion time equals to C_{max}. More specifically the first term of the equations of motion is equal to 0 except for the latter that is equal to $-\mu_{ms}$.

The starting time of each activity is calculated by that algorithm [27]. Briefly, according to this, each activity (operation) is assigned to a resource regard to its processing time. More specifically the shortest first unassigned activity is assigned to the first available resource that can execute it.

Let us consider a separate independent RNN aiming at the second objective function. So, we can perform the same calculations to find the equations of motion for the neural network with the energy function given in the equation (8). So, the equation of motion of the first neuron is as follows.

$$\frac{dS'_{11}}{dt} = -\mu_{wt}\frac{\partial E_{wt}(S)}{\partial S'_{11}}$$

Where S'_{11} is the starting time of the activity α_{11} regard to the Weighting Tardiness objective function. Let us calculate the complete formula of the first equation of motion:

$$\frac{dS'_{11}}{dt} = -\mu_{wt}\frac{\partial E_{wt}(S)}{\partial S'_{11}} = -\mu_{wt}\frac{\partial(\sum_{i=1}^{3} w_i T_i)}{\partial S'_{11}}$$
$$-\mu_{wt}K_{wt}\frac{\partial(\rho_2(c_1)+\rho_2(c_7)+\rho_2(c_8)+\rho_2(c_{23})+\rho_2(c_{24}))}{\partial S'_{11}}$$

The second term of the above equation is identical with the second term of the equation of motion of the first objective. So, we have to calculate the first term in order to have the exact form of the equation of motion as follows.

$$\frac{dS'_{11}}{dt} = -\mu_{wt}\frac{\partial(\sum_{i=1}^{3} w_i T_i)}{\partial S_{11}}$$
$$-\mu_{wt}K_{wt}[2(\delta_{11,23}+\delta_{11,31})S'_{11}$$
$$-S'_{12} + (1-2\delta_{11,23})S'_{23} + (1+\delta^2_{11,23}+\delta^2_{11,31})$$
$$\times\phi_{11} + (1-\delta^2_{11,23})\phi_{23} + (1-2\delta_{11,31})S'_{31}$$
$$+(1-\delta^2_{11,31})\phi_{31}]$$

More specifically, the first term of the above equation is as follows.

$$-\mu_{wt}\frac{\partial(\sum_{i=1}^{3} w_i T_i)}{\partial S'_{11}}$$
$$= -\mu_{wt}\frac{\partial(w_1T_1+w_2T_2+w_3T_3)}{\partial S_{11}}$$
$$= -\mu_{wt}\frac{\partial(max(0,S'_{13}+\phi_{13}-d_1)+2max(0,S'_{23}+\phi_{23}-d_2))}{\partial S'_{11}}$$
$$+\frac{\partial(3\times max(0,S'_{33}+\phi_{33}-d_3))}{\partial S'_{11}} = 0$$

So, the first equation of motion of the RNN that aiming at the Total Weighted Tardiness objective is similar to the one for the Makespan objective i.e.

$$\frac{dS'_{11}}{dt} = -\mu_{wt} K_{wt} [2(\delta_{11,23} + \delta_{11,31})S'_{11} - S'_{12} + (1 - 2\delta_{11,23})S'_{23} + (1 + \delta^2_{11,23} + \delta^2_{11,31})\phi_{11} + (1 - \delta^2_{11,23})\phi_{23} + (1 - 2\delta_{11,31})S'_{31} + +(1 - \delta^2_{11,31})\phi_{31}]$$

and the first neuron of the RNN is given by the following equation:

$$S'_{11}(t+1) = S'_{11}(t) + \frac{dS'_{11}}{dt} = S'_{11}(t) -\mu_{wt} K_{wt} [2(\delta_{11,23} + \delta_{11,31})S'_{11}(t) - S'_{12}(t) + (1 - 2\delta_{11,23})S'_{23}(t) + (1 + \delta^2_{11,23} + \delta^2_{11,31})\phi_{11} + (1 - \delta^2_{11,23})\phi_{23} + (1 - 2\delta_{11,31})S'_{31}(t) + (1 - \delta^2_{11,31})\phi_{31}$$

As we observe the equation of motion is not dependent on the objective function i.e. the Total Weighted Tardiness. Actually, the only neuron that is affected by the form of the objective function is the neuron which represents the last activity (operation) for every project (job). For instance, we calculate the equation of motion of the neuron represents the last activity of the first project i.e. S'_{13} similarly.

The second term- fraction of the correspondent equation is irrelevant to the selected objective function. On the other hand, the calculation of the first term-fraction leads to:

$$-\mu_{wt} \frac{\partial \max(0, S'_{13} + \phi_{13} - d_1)}{\partial S'_{13}} = 0$$
$$if\ S'_{13} + \phi_{13} - d_1 \leq 0$$
$$= -\mu_{wt}\ else\ S'_{13} + \phi_{13} - d_1 > 0$$

In conclusion, the second RNN aiming at the Total Weighted Tardiness objective has only one difference regard to the first RNN. The three neurons S'_{13}, S'_{23}, and S'_{33} have equation of motion that contains the additional term $-\mu_{wt}$.

V. Experimental Results and Discussion

There are two choices for the benchmarks that can be utilised. The first is the design and the implementation of a new set that can be considered as benchmarks and the second is the adoption of an established set in order to fit the MPM problem. The first option requires an enormous effort and the necessary recognition by the research community. Consequently, we have to adopt some JSSP benchmarks and exploit them in order to serve our goal which is the evaluation of the proposed techniques. The investigation and the final selection of this set was not a trivial procedure due the problem modelling.

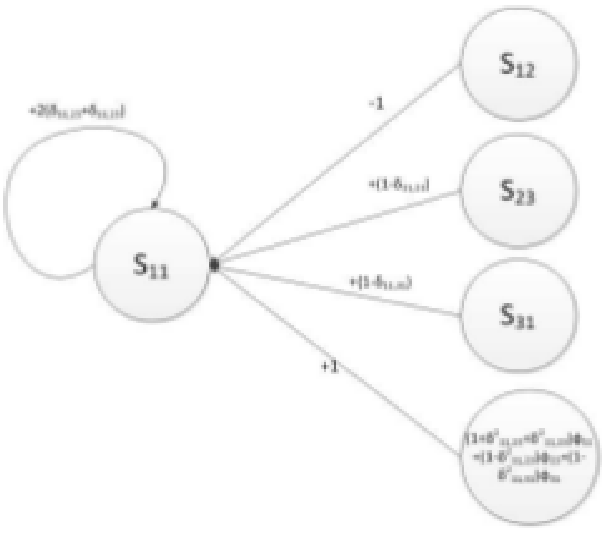

Figure 2. The first neuron of the example

This is actually a serious difficulty because most of the literature has concentrated on the original job shop scheduling problem. According to the formulation of JSSP each task of each job can be assigned to a unique resource. However, this is not the case for the project portfolio. The project teams usually are responsible for different activities of different projects and the project manager assigns the activities to them. The simple JSSP has been researched by many scholars during the last forty years and there are many benchmarks that have been designed and tested successfully. Unfortunately, the MPM form, which is generalisation of our problem, did not obtain an established set of problem instances that is globally accepted and can be utilised.

Very few researchers have designed and tested sets of benchmarks that are suitable for the specific MPM problem. One of them is B. Jurisch [1] who constructed a set of benchmarks via the extension of the well known benchmark instances created by Lawrence [16]. More specifically, the author changed the established benchmarks by adding machines (resources) randomly to the sets $r_{ix} \subseteq R$ that can execute an activity a_{ix}. So, he generated three groups of benchmarks (edata, rdata and vdata) that have as crucial properties the average and the maximum cardinality of the sets $r_{ix} \subseteq R$. The edata set contains instances that allow few choices for each activity (operation). In the rdata set instances the activities have a moderate number of possible resources to be assigned. Finally, the vdata set is the one that allows the most choices for each activity. The author presented tables with results using several exact algorithms and the optimal solutions (or their low bounds) for their benchmark instances. Lastly, many research efforts are based on these benchmarks [11], [28], [18].

We have to underline that the majority of the benchmark instances are designed for the first objective i.e. C_{max}. Few researchers tried to implement suitable benchmarks for the second one i.e. $\sum w_i T_i$. An interesting set has been proposed as an extension of the Lawrence [16] and Jurisch [1] benchmarks in order to accomodate the objective of the Tardiness [6], and it has been exploited for our model.

According to them, there are no benchmark instances available for job shop scheduling problem with due date. They also asserted that the established set of benchmarks proposes by other researchers have very few jobs and operations and they are limited. They also contended that their set of benchmarks are larger than previous sets and satisfy more objective functions. So they produce 600 problem instances focused on the makespan objective and 80 instances focused on Tardiness. Subsequent researchers exploited their method in order to obtain benchmarks with due date. Moreover, these researchers [6] tested the effectiveness of their proposed benchmarks using solution methods already implemented by others.

In conclusion, the research community did not adopted the set of benchmarks created by this work. However, many researchers adopted and adapted their method that generates randomly due date for their benchmark instances. We will make use of this work for the due dates by applying then to the established Jurisch [1] benchmarks.

Actually, the authors of some interesting works [28], [18] combined the data sets created by [1] and the method proposed by [6]. So they calculated the due date as follows:

$$d_i \hookrightarrow U[k \times (1 - \frac{D}{2}); k \times (1 + \frac{D}{2})] \qquad (9)$$

where $k = (1 + \frac{T \times N}{M}) \times \sum_{j=1}^{n_j} \phi_{i,j}$, N the number of the jobs, M the number of the resources, T the average number of tardy jobs (indicative values 0.3, 0.5), D due date range (indicative values 0.5, 2.5), and $\phi_{i,j}$ the processing time of the activity j of the project i.

According to the latter work, a set of benchmarks can be constructed by using the uniform probability distribution and the parameters T which is the expected number of the tardy jobs and R which is the due date range parameter, proposed by [6]. The selected values for those parameters are $T = 0.3$ and $R = 0.5$. The random values for the due time i.e. the deadlines are obtained from the distribution (9).

In Figures 3, 4 there are two graphs which present the evolution of the RNN regarding to the first objective function, the makespan. As we can see, after few iterations, the value of the makespan converges towards the known optimum or near optimum. The convergence point is higher than that optimum. Actually, this convergence point is achieved usually for $K_{ms} = 0.01$ and $\mu_{ms} = 0.5$ The interesting finding in these graphs is that for greater values of μ_{ms} ($\mu_{ms} > 0.5$) in most benchmark instances the RNN converges to lower values approaching the optimum. However, in that case there are apparent indications about potential singularities.

In table III, the results of the RNN evolution for the makespan are presented in comparison with the known optimum or near optimum for the set of benchmark instances $la01 - la10$ modified in order to fit to our model [1]. As one can easily observe this RNN converges very fast towards a solution that is significantly greater than the known optimum. Actually, the mean distance between the two solutions is approximately 13.4%. Apparently, the RNN's results are insufficient exclusively regarding to the makespan objective. Similarly, we have the same performance regarding to the second objective, the Total Weighted Tardiness. However, if we consider the scalar compound objective that comprises the makespan and the total weighted tardiness we have very promising findings.

The limitations of our approach are quite obvious. First, we have to investigate and amend the potential singularities that can be arose in every RNN approach. Secondly, we have to determine more accurate values for coefficients of the scalar objective. Third, we need to improve the RNN evolution by avoiding the possible local minima in order to converge towards lower points close to the global minimum. Finally, we have to explore an alternative approach, i.e. to design an alternative unified RNN that will include both objective functions.

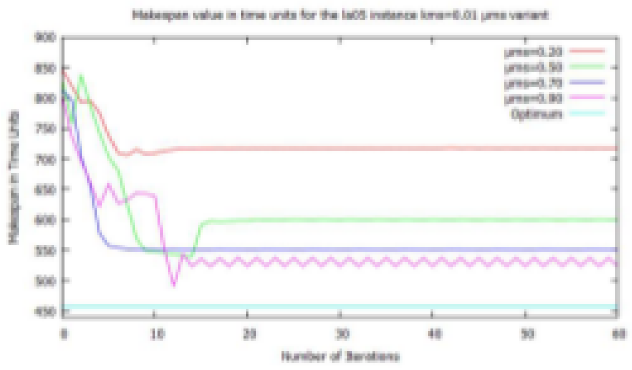

Figure 3. The results of the RNN evolution for the la05 instance

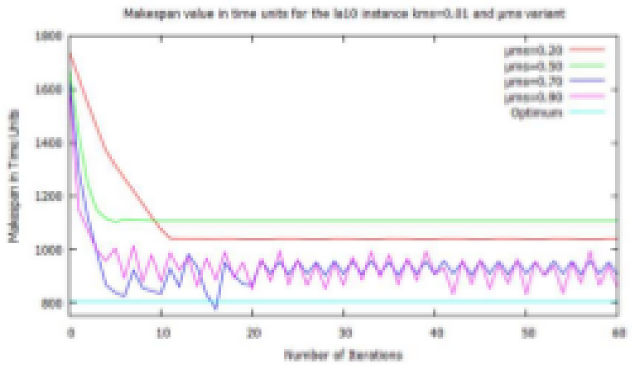

Figure 4. The results of the RNN evolution for the la10 instance

Table III
MAKESPAN MINIMUM VALUE FOR A SUBSET OF THE VDATA BENCHMARKS

Instance	*Optimal Makespan*	*RNN's result ($\mu_{ms} = 0.5$) of Makespan*	*% divergence*
la01	570	661	13.7
la02	529	807	34.4
la03	477	568	16.0
la04	502	630	20.3
la05	457	539	15.2
la06	799	852	6.2
la07	749	775	3.3
la08	765	758	-0.9
la09	853	840	-1.5
la10	804	1108	27.4

VI. Conclusions and Future Work

This paper is an extension of an ongoing research work and focuses on the multipurpose machines Job Shop Scheduling Problem. That case of JSSP can be utilized for the modelling of the project portfolio management. In this work we reviewed the literature, extended a bi-objective system model based on the job shop scheduling problem and modelled it as a combination of two separate recurrent neural networks. In addition, we provided an example within two RNNs regarding to Makespan and Total Weighted Tardiness objectives. Moreover, we present the promising findings of our approach using a set of well known benchmark instances and we provide the discussion about them and their singularity issues. One goal of the future work is to approach a near optimal solution via the adjustment of the described RNNs and the trade-off between the two adopted objective functions; the makespan and total weighted tardiness. Finally, we will apply and test the model on other sets of benchmark instances and we will study the potential singularities more extensively.

References

[1] Bernd Jurisch. *Scheduling Jobs in Shops with Multi-Purpose Machines*. PhD Thesis, University of Osnabrueck, 1992.

[2] Tyson R. Browning and Ali A. Yassine. Resource-constrained multi-project scheduling: Priority rule performance revisited. *International Journal of Production Economics*, 126(2):212–228, 2010.

[3] Peter Brucker, Andreas Drexl, Rolf Mohring, Klaus Neumann, and Erwin Pesch. Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112(1):3–41, January 1999.

[4] Peter Brucker and Sigrid Kunst. *Complex scheduling*. GOR-Publications. Springer, Berlin, 2006.

[5] Daniel Antonio Callegari and Ricardo Melo Bastos. Project management and software development processes: Integrating RUP and PMBOK. In *International Conference on Systems Engineering and Modeling, ICSEM '07*, pages 1–8, Haifa, 2007. IEEE.

[6] Demirkol, E., Mehta, S., and Uzsoy, R. Benchmarks for shop scheduling problems. *European Journal of Operational Research*, 109(1):137–141, 1998.

[7] G. Dreyfus. *Neural Networks Methodology and Applications*. Springer Berlin Heidelberg, Heidelberg, 2005.

[8] Simon Haykin. *Neural Networks- A Comprehensive Foundation*. Prentice-Hall Inc., New Jersey, second edition.

[9] Christian Heimerl and Rainer Kolisch. Scheduling and staffing multiple projects with a multi-skilled workforce. *OR Spectrum*, 32(2):343–368, 2010.

[10] Bob Hughes and Hogan Cotterel. *Software Project Management*. McGraw- Hill Education, Berksire UK, fifth edition, 2009.

[11] Johann Hurink, Bernd Jurisch, and Monika Thole. Tabu search for the job-shop scheduling problem with multi-purpose machines. *OR Spektrum*, 15:205–215, 1994.

[12] D.F. Jones, S.K. Mirrazati, and M. Tamiz. Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, 137(1):1–9, 2002.

[13] M-Tahar Kechadi, K.S Low, and G. Congalves. Recurrent neural network approach for cyclic job shop scheduling problem. *Journal of Manufactoring Systems- Elsevier*, (In press), 2012.

[14] Majid Khalili and Reza S. Tavakkoli-Moghaddam. A multi-objective electromagnetism algorithm for a bi-objective flowshop scheduling problem. *Journal of Manufacturing Systems*, 31(2):232–239, 2012.

[15] Zohar Laslo. Project portfolio management: An integrated method for resource planning and scheduling to minimize planning/scheduling-dependent expenses. *International Journal of Project Management*, 28:609–618, 2010.

[16] S. Lawrence. *Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (Supplement)*. Graduate School of Industrial Administration,Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1994.

[17] Office of Government Commerce. *Managing successful projects with PRINCE2*. TSO, London, fourth edition, 2005.

[18] Pisut Pongchairerks. Particle swarm optimization algorithm applied to scheduling problems. *ScienceAsia*, 35:89–94, 2009.

[19] PMI. *A Guide to the Project Management Body of Knowledge*. Project Management Institution, Pennsylvania, fourth edition, 2008.

[20] Christoph Schwindt. *Project management and resource allocation*. Springer-Verlag, Heidelberg, 2005.

[21] A. Senouci and H.R. Al-Derham. Genetic algorithm-based multi-objective model for scheduling of linear construction projects. *Advances in Engineering Software*, 39(12):1023–1028, 2008.

[22] Kate A. Smith. Neural networks for combinatorial optimization: A review of more than a decade of research. *INFORMS Journal on Computing*, 11(1):15–34, 1999.

[23] J.C. Tay and N.B. Ho. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. *Computers and Industrial Engineering*, 54(3):453–473, 2008.

[24] Williems T.M. and Rooda J.E. Neural networks for Job-Shop scheduling. *Control Eng. Practice*, 2(1):31–39, 1994.

[25] Mariem Trojet, Fehmi H'Mida, and Pierre Lopez. Project scheduling under resource constraints: Application of the cumulative global constraint in a decision support framework. *Computers & Industrial Engineering*, 2010.

[26] Dimitrios Tselios, Ilias K. Savvas, and Tahar MT Kechadi. Project portfolio: A job scheduling approach. In *Sixth International Conference on Complex, Intelligent, and Software Intensive Systems*, pages 847–852, Palermo, Italy, July 2012. IEEE.

[27] Dimitrios Tselios, Ilias K. Savvas, and Tahar MT Kechadi. The weighted tardiness as objective function of a RNN model for the job scheduling problem. In *European Modelling Symposium EMS2012 (EMS2012)*, pages 13–18, Malta, Malta, November 2012. IEEE.

[28] Vilcot, G. and Billaut, J.-C. A tabu search algorithm for solving a multicriteria flexible job shop scheduling problem. *International Journal of Production Research*, 49(23):6963–6980, December 2011.

[29] Lucas Wiedemann. *Introduction in IT Project Management*. Lucas Wiedemann (Standard Copyright License), first edition, March 2010.

[30] L.-N. Xing, Y.-W. Chen, and K.-W. Yang. Multi-objective flexible job shop schedule: Design and evaluation by simulation modeling. *Applied Soft Computing Journal*, 9(1):362–376, 2009.

[31] Rui Zhang, ShiJi Song, and Cheng Wu. A hybrid artificial bee colony algorithm for the job shop scheduling problem. *International Journal of Production Economics*, 141(1):167–178, 2013.