Machine Learning Heuristic for Solving Multi-Mode Resource-Constrained Project Scheduling Problems

Patience I. Adamu, Micheal C. Agarana, and Hilary I. Okagbue,

***Abstract*— The non-preemptive resource-constrained project scheduling problem is considered in this work. It is assumed that each activity has many ways of execution and the objective is to find a schedule that minimizes the project's completion time (multi-mode RCPSP). Methods that are based on priority rules do not always give the needed very good results when used to solve multi-mode RCPSP. In solving large real-life problems quickly though, these methods are absolutely necessary. Hence good methods based on priority rules to get the primary results for metaheuristic algorithms are needed. This work presents a novel method based on priority rules to calculate the primary solutions for metaheuristic algorithms. It is a machine learning approach. This algorithm first of all uses Preprocessing to reduce the project data in order to speed up the process. It then employs a mode assignment procedure to obtain the mode of each job. After which the algorithm uses machine learning priority rule to get the precedence feasible activity list of the project's tasks. Finally, it then uses the Serial Schedule Generation Scheme to get the total completion time of the project. In our experiments, we use our algorithm to solve some problems in the literature that was solved with metaheuristic procedures. We compared our results with the initial solutions the authors started with, and our results competes favorably with the initial solutions, making our algorithm a good entry point for metaheuristic procedures.**

***Index Terms*—machine learning, project scheduling, resource constraints, metaheuristic procedures.**

I. INTRODUCTION

THE NON-preemptive multi-mode RCPSP finds the minimum schedule which minimizes the project's completion, while satisfying the precedence and the resource constraints. The precedence constraint ensures that the execution of a task cannot start until all its direct predecessors are completely executed, while the resource constraint ensures that the available resources, every period are not exceeded. The different ways called modes of performing each job is considered, and each job is usually performed in one of its modes. [2]. Three types of resources are used, the renewable, non-renewable and doubly constrained resources [13]. The renewable resources are made available per-period (e. g. the number of skilled workers needed daily) and the non-renewable resources are made available for the entire project (for example the total money needed for the project). Doubly constrained resources are limited both for each period and for the total project, but are not considered separately. This is because when the sets of the renewable and non-renewable resources are enlarged the doubly constrained resources are incorporated.

This problem has found application in many real life applications and industries, such as project management and crew scheduling, construction engineering, production planning and scheduling, fleet management, machine assignment, automobile industry and software development.

However, this problem has been shown to be an NP-hard optimization problem (Kolisch, [5]) if the number of non-renewable resources is more than one. Hence exact methods become intractable as the number of activities and the number of modes for each activity increases [10]. Therefore, large – real world problems of this class can only be tackled by heuristic.

Heuristic algorithms based on priority rule have been one of the most important solution techniques for solving the single mode case of this problem and acceptable results has been obtained even for large sized projects.

However, solving multi-mode RCPSP with methods based on priority rules, do not give the desired results always, but they are very essential when solving real-life problems quickly. Therefore, good methods based on priority rules are needed to calculate the primary solutions for metaheuristic procedures such as Simulated Annealing (SA) and Genetic Algorithms (GA) [Mohammed et. al]. This is the motivation of this study.

Methods based on priority rules for solving multi-mode RCPSP combine scheduling generation schemes (SGS), priority rules and mode assignment rules to construct specific algorithms. Two different SGS are available: the serial SGS and the parallel SGS. Both schemes use a step-by-step process to build schedules. A priority rule is used to sort out conflicts between jobs competing for resources when constructing the activity list and a mode assignment method is used to allocate an execution mode to each job, thereby determining its duration and the resources needed.

Many heuristic algorithms exist for solving multi-mode RCPSP. Examples are Rezaeian et al [9], Jarboui et al. [4],

Manuscript received December 21, 2017; revised Jan 15, 2018.
P. I. Adamu is a faculty in the Department of Mathematics, Covenant University, Ota, Nigeria. patience.adamu@covenantuniversity.edu.ng
M. C. Agarana is a faculty in the Department of Mathematics, Covenant University, Ota, Nigeria.
michael.agarana@covenantuniversity.edu.ng
H. I. Okagbue is a faculty in the Department of Mathematics, Covenant University, Ota, Nigeria. hilary.okagbue@covenantuniversity.edu.ng

ISBN: 978-988-14047-8-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Hartman [3], Kolisch and Drexl [6]. Some of these approaches use priority rule based scheduling technique to construct feasible schedules.

In the literature, there are quite a number of priority rules available, and selecting the best one for a particular input problem is extremely difficult.

This work presents a machine learning algorithm for solving the multi-mode RCPSP. Our approach finds a schedule that minimizes its completion time by implementing the following steps (Fig. 4):

Step 1: Modifies the data of the project by preprocessing in order to reduce the search space.

Step 2: Uses a local search method to find a mode assignment for the jobs of the project that is feasible.

Step 3: Uses machine learning priority rule (MLPR) to find the activity list of the jobs.

Step 4: Finds a schedule that is feasible by using serial schedule generation scheme

In our experiment we use our algorithm to find initial solutions for some metaheuristic procedures and our solutions outperform the author's initial solutions for the metaheuristic procedures: We solve the project example problem of Hartmann [3], we got 10 units, while the author's initial solution for his Genetic algorithm (GA) was 15 units.

Also, Mohammad et. al. [8] presented a GA for solving the multi-mode RCPSP. The initial solution to their project example was 18 units as an *entry point*, before it improved it using Multi- mode forward-backward iteration (MM-FBI) method. Our algorithm, which may be used as an *entry point*, when used to solve the same problem got 12 units.

The main contributions of this paper are:

1) To present an improved local search for mode assignment.
2) To propose a novel heuristic for solving multi-mode resource-constrained project scheduling problems.

II. PRELIMINARIES AND RELATED WORKS

A. *Types of Resources for the multi-mode RCPSP*

The resources needed to execute the jobs of projects was categorized into three (Slowinski [13] and Weglarz [15]). The renewable, non-renewable and the doubly constrained resources:

(i) *Renewable Resources:* These are resources that are made available per period (hourly, daily, weekly, monthly, etc) or each period (say, every day) the quantity of each resource (say, manpower) made available is assumed constant. For example, a company can decide to make available 5 skilled and 8 unskilled laborers on a daily basis for the execution of a particular project until it is completed. That is, the quantity of each resource is renewed every period.

(ii) *Nonrenewable Resources:* These are resources whose availabilities are not limited per period, but on the total project. For example, an individual may decide that the building project must not be more than a particular amount of money (say, X dollars). That means that money to be spent is limited on the total project basis. This is the budget for the project. Again the individual may decide to use a total 50 skilled and 80 unskilled laborers for the total project. This implies that the total availability of manpower is also limited.

(iii) *Doubly Constrained-Resources:* These are resources that are limited both on total project basis and on per period basis. Money falls into this category, because, apart from having a budget for a project, the per-period cash flows are also limited. If the renewable and non-renewable resources are enlarged appropriately, the doubly constrained resources are already considered. Hence, in the model only the renewable and non-renewable resources are explicitly considered.

The Description of multi-mode RCPSP

Considered is a project with J number of jobs. Precedence relations exist between jobs. That is, a job is not started until all its direct predecessors are completed. Activity-On-Node (AON) network is used to describe the structure of the project. The nodes of the network represent the jobs and the arcs represent the precedence relations that exist between jobs. Job 1 and job N are project start and project finish jobs respectively. If the**y** are not naturally available in a problem, dummy start job and dummy end job are attached to the problem. Different ways (called modes) of executing a job is considered, and executed in one out of its modes. For example, in a building construction, blocks (bricks) can be obtained: by buying the already made blocks (bricks) from the block industries (mode 1, m_1) or by buying the materials and have the blocks molded (mode 2, m_2). These two methods of getting blocks have different levels of resource requirements and different durations. Jobs started in a particular mode are completed in that same mode without interruption. Executing a job in a mode requires a set of renewable resources, a set of non-renewable resources and a set of doubly constrained resources. The objective is to find a minimum and feasible schedule with respect to the constraints imposed by the precedence relations and the limited resource availability to minimize project completion time. A description of different versions of this problem and the integer programming formulations can be seen in different papers: Talbot (14), Hartmann [3], Sprecher et. al.(11).

$$Min \sum_{m=1}^{M_j} \sum_{t=EF_j}^{LF_j} t.x_{jmt} \quad (1)$$

$$\sum_{m=1}^{M_j} \sum_{t=EF_j}^{LF_j} x_{jmt} = 1;\ j = 1, \cdots, J \quad (2)$$

$$\sum_{m=1}^{M_h} \sum_{t=EF_h}^{LF_h} t.x_{hmt} \leq \sum_{m=1}^{M_j} \sum_{t=EF_j}^{LF_j} (t - d_{jm}) x_{jmt};\ j = 2, \cdots, J;\ h \epsilon P_j \quad (3)$$

$$\sum_{j=1}^{J} \sum_{m=1}^{M_j} \sum_{q=t}^{t+d_{jm}-1} r_{jmk}\, x_{jmq} \leq R_{kt};\ k \epsilon K;\ t = 1, \cdots, T \quad (4)$$

$$\sum_{j=1}^{J} \sum_{m=1}^{M_j} \sum_{t=EF_j}^{LF_j} w_{jmi}\, x_{jmt} \leq W_i;\ i \epsilon I \quad (5)$$

$$x_{jmt} \epsilon [0,1]; j = 1, \cdots, J; m = 1, \cdots, m_j; t = EF_j, \cdots, LF_j \quad (6)$$

where j is the job index; j =1, . . ., J (J is the number of jobs in a project)

m is the mode index; m = 1, . . ., m_j (m_j is the number of possible modes for activity j)

t is the time index; t = EF_j, . . . , LF_j (EF_j is the earliest finish time and LF_j is the latest finish time of job j).

W_{jm} is the nonrenewable resource demand of job j in mode m,

ISBN: 978-988-14047-8-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

K is the number of renewable resources
I the number of non-renewable resources.
R_{kt} The level of available renewable resource k in time t.
W_i the total available budget of non-renewable resource i
The objective function (1) minimizes the project's completion time. Constraint (2) ensures that each job is assigned exactly one mode and one completion time Constraint (3) takes care of the precedence relations between jobs, while (4) guarantees that the non-renewable resource availability is not exceeded. This problem is NP-complete (Kolisch and Drexl [6]) and real world problems as the one we are tackling now can only be solved using heuristics.

B. Preprocessing

The basic idea of preprocessing is the removal of modes and/or nonrenewable resources from the data of the project in order to reduce the space of solution search. This reduction method was introduced by Sprecher et al. [12].

Reduction of the input data

The reduction of the input data uses the *bounding rule* which uses the following:
A mode cannot be used if it makes any schedule not to be feasible as far as renewable or non-renewable resources are concerned.
A mode of a job is not good enough to be used if another mode of that same job has a smaller length of period and smaller demands of resources
A non-renewable resource is not useful if the sum of the maximal demands of the jobs for this resource is not more than its availability.
Sprecher et al. [11], [12] found out that optimality is not lost when these modes and nonrenewable resources are removed and also that there are interaction effects with their removal. . Hence they proposed the following steps for reducing the data of a project:
Step 1: Remove all modes from the project data that makes any schedule not to be feasible.
Step 2: Remove all the non-renewable resource that are not useful.
Step 3: Expunge all modes that are not good enough.
Step 4: In step 3, if any mode is removed, go to Step 2.
Fig. 1 explains the algorithm that implements this bounding rule.

C. Machine Learning Priority Rule

The machine learning priority rule (Adamu and Aromolaran [1]) for the resource-constrained Project Scheduling Problem assembles a set of priority rules, and uses machine learning strategies to choose the one with the best performance at every point in time to construct an activity list of a project tasks. The one with better performance is used most frequently. This removes the problem of manually searching for the best priority rule amongst the dozens of rules that are available. Fig. 3 explains the implementation of this algorithm.

III. THE MACHINE LEARNING HEURISTIC (MLH)

Our machine learning approach, first of all reduces the search space of the input data by preprocessing.

Then a simple local search algorithm is used to assign a mode to each job. With the feasible mode assignment, the multi-mode RCPSP becomes single-mode RCPSP. To find the activity list of the jobs, MLH uses our machine learning priority rule (Adamu and Aromolaran [1]). Finally, serial schedule generation scheme is used to get the feasible schedule.

a) The algorithm reduces the search space of the input data:

Algorithm 1: Project Data Reduction Algorithm

Input: a) A project with J number of jobs ($j = \{1, 2, \cdots, J\}$).
b) The duration of each of the jobs.
c) The renewable resource demand of all the modes of each job.
d) The nonrenewable resource demand of all the modes of each job.
Output: Reduced project data.
(1) Remove all the *non-executable* modes of all renewable resources
(2) Remove all the *non-executable* modes with respect to all non-renewable resources.
(3) Remove all the redundant nonrenewable resources
(4) Remove all the inefficient modes
(5) Go to step 3
(6) Stop, if no mode is removed from step 4
(7) Go to Algorithm 2

Fig. 1: Algorithm to reduce the search space of the project data.

b) Mode Assignment Algorithm

The local search procedure converts a multi-mode case to a single-mode case. The procedure is used to generate a feasible mode assignment m = {m_1, m_2, . . ., m_n} for all the jobs, J = {j_1, j_2, . . ., j_n}.
The local search by Hartmann, [3] generates its first mode assignment randomly, while our local search generates its first mode assignment by assigning to each job, its mode that has the least renewable resource. And if a tie exists, then the mode with the minimum duration is chosen. So our local search is as follows:
A mode is first generated for each job in the project. Then the sum of their nonrenewable resource requirements, (W_{jm}) is calculated for each nonrenewable resource type (i =1, 2, , , , I). If this sum exceeds the available nonrenewable resource level (W_i, i =1, 2, , , , I), it implies that the mode assignment is infeasible and a simple mode assignment strategy is employed to improve the current mode assignment: A job is randomly selected and a mode with its next minimal renewable resource, different from its current mode. This leads to a another mode assignment m`. Then a check is made, to see if the sum of their nonrenewable resource requirements exceeds the available nonrenewable resource level (W_i, i =1, 2, , , , I). This process is continuously repeated until the mode assignment becomes feasible, which makes the multi-mode case to become a single-mode case.

Algorithm 2: Mode Assignment Algorithm

Input: Reduced project data from Algorithm 1
(1) Generate a mode assignment m by assigning to each job, j = 1,…,J. its mode with minimum renewable

ISBN: 978-988-14047-8-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

resource and if a tie exists, the job that has the least duration is chosen.

(2) Check the mode assignment for nonrenewable resource feasibility.

(3) If the mode assignment is infeasible, randomly select a job and change its mode, leading to a new mode assignment m'

(4) Go to step 2

(5) Stop if mode assignment is feasible, making the problem a Single-Mode case.

(6) Go to Algorithm 3

Fig.2: Algorithm to obtain the mode of every activity.

c) Obtaining the activity list

Algorithm 3: Machine Learning Algorithm to obtain the activity list

Input: a) A project with J number of jobs ($j = \{1, 2, \cdots, J\}$).

b) A set of *n* priority rules, $pr_1, pr_2, pr_3, \cdots, pr_n \ (i = 1, 2, \cdots, n)$.

Output: A feasible schedule of the Project's (completion time).

Require:

Let $P = \{p_1, p_2, p_3, \cdots, p_n\}$ be a set of probabilities where p_i is the probability of choosing a priority rule. Initialize $p_i(t = 1) = \frac{1}{n} \ \forall$ *the priority rules*.

i) Initial weight of each priority rule is one ($w_i(1) = 1$).

ii) Initial cost of each priority rule is assumed to be one ($c_i(1) = 1$)..

iii) Let AL = $\{j_1, j_2, \cdots, j_r\}; r < J$ be the activity list under construction.

for Step t = 1, 2, … ***do***

1) Choose a priority rule $pr_i(t)$ using P
2) Write out the set of Eligible activities.
3) Run the chosen rule $pr_i(t)$ to get the appropriate job to be scheduled.
4) Put the job j_r into the activity list AL that is being constructed.
5) Get the reward of the chosen priority rule.
6) Get the cost
7) Update success rates
8) Update costs
9) Update Probabilities of all the priority rules.

- If the activity list is finally got, use Serial Schedule Generation Scheme (SSGS) to construct a feasible schedule of the activities which gives the project's time of completion.

Fig.3: Algorithm to get the activity listing of the project tasks

d) Algorithm 4 summarizes all the steps of MLH for multi-mode RCPSP

Algorithm 4: Machine Learning Heuristic (MLH) for Multi-Mode RCPSP

Input: a) A project with J jobs in number ($j = \{1, 2, \cdots, J\}$).

b) A set made up of *n* priority rules, $pr_1, pr_2, pr_3, \cdots, pr_n \ (i = 1, 2, \cdots, n)$.

1) Reduce the search space by using Algorithm 1
2) Obtain a feasible mode assignment for the project's tasks using Algorithm 2
3) Get the activity Listing of the project's tasks by using Algorithm 3
4) Use the Serial Schedule Generation Scheme (SSGS) to construct a schedule that is feasible.
5) Get the project completion time (makespan) from the feasible schedule

Fig.4: A machine Learning Algorithm for the multi-mode RCPSP

IV. EXPERIMENTS

Experiment 1:

Experimental Design:

The project example of Hartmann 1997 has six jobs, each job has two modes, the type of renewable resource is one and type of nonrenewable resource type is one. The detailed information is presented in Fig. 1 and Table 1 below.

In Table Ren. RD means Renewable Resource Demand and NRen. RD means Nonrenewable Resource Demand.

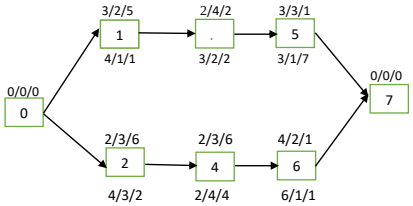

Fig. 1 : Project Example of Hartmann

Table 1: Explains the Project example of Fig.1

Job No.	Modes	Duration	Ren. RD	NRen. RD
0	1	0	0	0
1	1	3	2	5
	2	4	1	1
2	1	2	3	6
	2	4	3	2
3	1	2	4	2
	2	3	2	2
4	1	2	3	6
	2	2	4	4
5	1	3	3	1
	2	3	1	7
6	1	4	2	1
	2	6	1	1
7	1	0	0	0
Available Resources			4	15

Results:

a) Using our local search for mode assignment we got Table 2.

Table 2: Project Example after Mode Assignment

Job No.	Modes	Duration	Ren. RD
0	1	0	0
1	2	4	1
2	1	2	3
3	2	3	2
4	1	2	3
5	2	3	1
6	2	6	1
7	1	0	0
Available Resources			4

b) Solving this problem with our algorithm we got the activity list of {2:1, 4:1, 6:2, 1:2, 3:2, 5:2} with the completion time for the project to be 10 units. 2:1 means job 2 modes 1.

ISBN: 978-988-14047-8-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

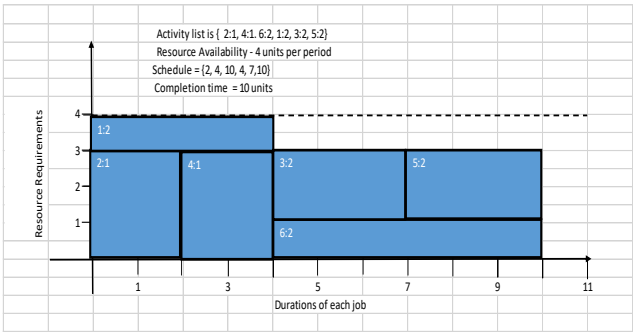

Fig. 2: The Project Example Schedule

In Hartmann [3], the initial completion time (*the entry point*) for this example before using GA was 15 units, but we got 10 units as an *entry point* with our algorithm.

Experiment 2:

Mohammad et. al. [8] presented a genetic algorithm (GA) which uses random key representation to solve the multi-mode RCPSP.

They considered a project with 9 jobs, one type of resource that is renewable and another type of resource that is non-renewable, available at level two and twenty-eight, respectively. Fig. 3 and Table 3 give complete information of this project.

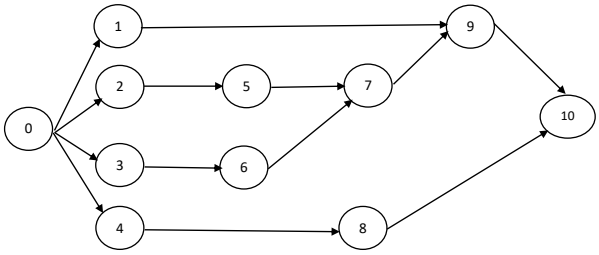

Fig 3: Activity network of the project example of Mohammad et. al

Table 3: Project Example of Mohammad et. al

Job No.	Mode	Duration	Ren. RD	NRen. RD
0	0	0	0	0
1	1	2	2	4
	2	3	1	2
2	1	2	1	5
	2	4	1	2
3	1	2	1	3
	2	3	1	1
4	1	1	1	2
	2	2	1	1
5	1	2	2	5
	2	4	2	3
6	1	2	1	4
	2	3	1	3
7	1	1	2	2
	2	2	2	1
8	1	1	2	3
	2	3	1	2
9	1	4	1	3
	2	6	1	1
10	0	0	0	0

Result:

a) Using our local search for the mode assignment we got Table 4:

Table 4: Project Example of Mohammad et. al after mode assignment

Job No.	Mode	Duration	Ren. RD
0	0	0	0
1	2	3	1
2	1	2	1
3	1	2	1
4	1	1	1
5	1	2	2
6	1	2	1
7	1	1	2
8	2	3	1
9	1	4	1
10	0	0	0
Available Resources			2

b) Solving this problem with our algorithm we got the activity list to be {4:1, 3:1, 2:1, 1:2, 6:1, 5:1, 7:1, 8:2, 9:1} and the project's time of completion is 12 units.

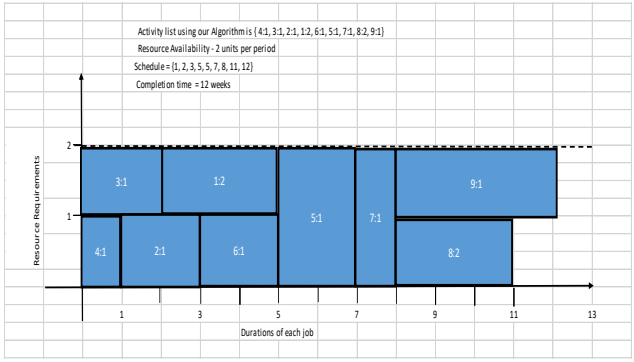

Fig. 4: Project Schedule of Mohammad 's Project Example

In Mohammad et al. [8], the initial completion time (*the entry point*) for this example before improving it with Multi- mode forward-backward iteration method and then use GA was 18 units, but we got 12 units as *entry point* with our algorithm.

V. CONCUSION

This paper examines an intelligent way of determining *good* initial solutions for metaheuristic procedures. It uses machine learning approach. Basically, this approach assembles a number of priority rules and automatically determines the best one to use at a given time when building a feasible activity list of a multi-mode resource - constrained scheduling project. It rapidly learns and uses the priority rule that is best amongst the assembled priority rules which use a trade-off between success rate and cost. We considered some experiments to confirm that our approach may be used as an *entry point* for metaheuristic procedures, like Genetic Algorithm (GA).

ACKNOWLEDGMENT

The financial support from Covenant University is appreciated.

REFERENCES

[1] P. I. Adamu, O. T. Aromolaran; "Machine Learning Priority Rule (MLPR) for Solving Resource-Constrained Project Scheduling Problems," In Proc. Int. Conf. on Machine Learning and Data Analysis, pp. to appear, an Francisco, USA, October, 2018.

[2] S.E. Elmaghraby, Activity Networks: Project planning and Control by Network Models. Wiley, New York, 1977.

[3] S. Hartmann, *Project scheduling with multiple modes: A genetic algorithm*, Manuskripte aus den Instituten fur Betriebswirtschaftslehre der Universit at Kiel, no. 435, 1997

[4] B. Jarboui, N. Damak, P. Siarry, A, Rabai, "Combinatorial Particle Swarm optimization for Solving multi-mode resource-constrained project scheduling problems" in *Applied Mathematics and Computation*, Vol 195, Issue 1, 299-308, 2008,
[5] R. Kolisch, Project-Scheduling under Resource-constraints: Efficient heuristics for several problem classes. Physica-Verlag, Hiedelberg; 1995.
[6] R. Kolisch and A. Drexl, Adaptive Search for Solving Hard Project Scheduling Problems, *Naval research Logistics*, vol. 43, pp. 23-40,1996.
[7] R. Kolisch, and A. Drexl, **"**Local search for nonpreemptive multi-mode resource-constrained project scheduling". *IIE Transactions*, vol. 29: pp. 987-999, 1997.
[8] S. H. Mohammad, R.A. Mohammad and A. Yagub; An efficient genetic algorithm for solving the multi-mode resource-constrained project scheduling problem based on random key representation.; *International Journal of Supply and Operations Management;* Volume 2, Issue 3, pp. 905-924, November 2015.
[9] J. Rezaeian, F. Soleimani, S. Mohaselafshary, A. Arab, "Using a meta-heuristic algorithm for solving the multi-mode resource-constrained project scheduling problem.International Journal of Operational Research," vol 24, issue 1, pp. 1-16, 2015
[10] A. Sprecher and A. Drexl, "Multi-Mode Resource-Constrained Project Scheduling Problems by A Simple General and Powerful Sequencing Algorithm," European Journal of Operational Research. vol. 107, pp. 431-450. 1998.
[11] A. Sprecher, S. Hartmann, and A. Drexl, "Project scheduling with discrete time-resource and resource-resource tradeoffs", Research Report 357, Institut fir Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel, 1995.
[12] A. Sprecher, S. Hartmann, and A. Drexl, "An Exact Algorithm for Projects Scheduling with Multiple Modes," *OR Spektrum*, vol. 19, pp. 195-203, 1997.
[13] R. Slowinski, "Two Approaches to Problem of Resource Allocation Among project Activities: A Comparative Study," Journal of the Operational Research Society. (31): 711-723, 1980.
[14] Talbot, F.B., "Resource-Constrained Project Scheduling with Time – Resource Trade-Offs: The Nonpreemptive Case," *Management Science*, vol. 28, no. 10, pp. 1197-1210. 1982.
[15] J. Weglarz, "On Certain models of resource allocation problems", *Kybernetics*, vol. 9, pp. 61-66, 1980

ISBN: 978-988-14047-8-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)