

A NEURAL NETWORK FOR RESOURCE CONSTRAINED PROJECT SCHEDULING PROGRAMMING

Ömer ÖZKAN^a, Ümit GÜLÇİÇEK^b

^a*Istanbul Medeniyet University, Department of Civil Engineering, 34730, Istanbul, Turkey*

^b*Kahramanmaraş Sutcu Imam University, Elbistan Vocational School, 46300, Kahramanmaraş, Turkey*

Received 28 Apr 2012; accepted 05 Nov 2012

Abstract. The resource constrained project-scheduling problem (RCPS) aims to minimize the duration of a project. RCPS is prevalently used in programming the projects with high number of activities and resources such as construction projects. In this study, 240 projects such as residential, office, school, etc. are designed and programmed under limited resources. The resource amounts of these projects are determined using three priority rules, these are Latest Finish Time, Minimum Slack Time and Maximum Remaining Path Length which have the highest performance according to the literature, in the amounts of 2, 4, 6 and 8. The project times are estimated using artificial neural network (ANN). A correlation coefficient of 0.70 was obtained from the ANN estimation model.

Keywords: project management, construction project, resource constraints, priority rules, artificial neural network.

Introduction

The optimum allocation of restricted resources over time is the principal concern of project scheduling, which thereby handles the task of defining the set of activities to be carried out at a particular point in time; in addition to playing a significant role in various fields of engineering (Idoro 2012). Programming large scale projects requires the deployment of analytical solutions for the purposes of scheduling activities and resource allocation. The associated tools are collectively denominated as the Critical Path Method (CPM). The Critical Path Method is used by practitioners for the purpose of planning and controlling large-scale projects especially in industries like manufacturing and construction. As the most widely employed project scheduling method in these sectors, CPM is primarily focused on the aspect of time. A CPM analysis traditionally operates under the assumption of unlimited resources. However, during the course of the scheduling process of a real project, the reality of constrained resources must be taken into account; such as limited crew sizes, equipment levels, and materials (Leu *et al.* 1999; Rogalska, Hejducki 2007).

RCPS involves assigning jobs or tasks to a resource or set of resources with limited capacity in order to meet some predefined objective (Yang *et al.* 2001). The objective of RCPS is to minimize the duration or total cost of project under the assumption of constrained resources (Bruni *et al.* 2011). The method of RCPS is commonly practiced in a variety of industrial fields such

as engineering, architectural design, manufacturing, software development, and project management (Brucker, Drexl 1999). A high level of complexity of a project schedule implies a high level of interconnection between the activities. Alternatively, a project is called to be complex in case the resources used in the project are widely spread (Nassar, Hegab 2006). Construction projects with resource constraints generally demand a large chunk of time and a well-organized operation to be completed, in addition to the utilization of a variety of resources and making a large amount of capital investment. The major goal of RCPS is to conduct problem analysis as well as evaluating the timetable alternatives and hence solve for the optimum one (Chassiakos, Sakellaropoulos 2005).

One of the scheduling methods used in resource constrained project scheduling problems is the heuristic method. Although it may not be possible to obtain an optimal solution with heuristic algorithms, they are still useful for solving the large-scale problems in addition to being easily run. In a sense, the heuristic method can be defined as a means of facilitating the process of reaching the optimal solution group through following a simple rule (Demeulemeester, Herroelen 1992). It focuses on two basic concepts; the first one is minimizing the time (Slowinski 1980; Talbot 1982; Jaśkowski, Sobotka 2004), and the second one is minimizing the project cost (Padman, Smith-Daniels 1993; Yang *et al.* 1993). The researchers have defined priority rules (solution algorithms) which minimize project cost and time

(Yang *et al.* 1993; Ulusoy, Ozdamar 1989). The heuristic method performance has been commonly measured in the literature through comparison of these priority rules (Ozdamar, Ulusoy 1996). Kolisch and Hartmann (1998) summarized and categorized heuristic priority rules, such as (Kolish, Hartman 1998):

- Maximum Remaining Path Length (MRPL);
- Latest Start Time (LST) LS_{ij} ;
- Latest Finish Time (LFT) LF_{ij} ;
- Shortest Processing Time (SPT) d_{ij} ;
- Most Total Successors (MTS) F_{ij} ;
- Earliest Start Time (EST) ES_{ij} ;
- Minimum Slack Time (MNSLCK) $LS_{ij} - ES_{ij}$.

In this study, 240 projects such as residential, office, school, etc. are designed and programmed under limited resources. The resource amounts of these projects are determined using 3 priority rules, these are LFT, MNSLCK and MRPL which have the highest performance according to the literature (Leu *et al.* 1999; Kolish, Hartman 1998), in the amounts of 2, 4, 6 and 8. The project times are estimated using the priority rules. Then the projects are classified based on the number of resources, resource amount, project width, number of transactions, manufacturing type etc. The obtained data will be entered into the artificial neural network (ANN) database to develop training and testing set of ANN models.

1. Algorithm and project description

1.1. Algorithm and priority rules

Heuristic algorithms leading to solutions by simple rules are used in project scheduling which especially in the construction investments. The priority rules used in resource-constrained conditions give various results depending on the size of the project, the number of resources and quantity of limitation. The performances of heuristic methods were tested and performances for limited resources and unlimited resources were evaluated. These pre-selected priority rules are MRPL (Maximum Remaining Path Length), LFT (Latest Finish Time) and MNSLCK (Minimum Slack Time). For each project, the priority rule that minimizes the project duration was determined. Project programming consists of two steps.

The first step in the algorithm is used for unlimited project planning models. The earliest start time (ES_{ij}), the latest start time (LS_{ij}), the earliest finish time (EF_{ij}), the latest finish time (LF_{ij}) and the slack ($S_{ij} = LS_{ij} - ES_{ij}$) of every activity are determined. i , start joint of the activity and j finish joint of the activity. In the second step, the resource constrained project is analyzed according to the priority rules. The program takes into account the daily resource requirements and checks each activity's resource requirement on a daily basis. The working principle of priority rules is described below:

- a) MRPL is an abbreviation for "Maximum Remaining Path Length". The time between the start and end

of a process is taken into consideration. This time for each process is calculated with the calculation below:

$$MRPL = T_p - T_i - S_{ij}, \quad (1)$$

where: T_p projects total time; T_i activity start time; S_{ij} slack.

MRPL tries to complete the longest activity at earliest to minimize the time overrun. If two activities are competing for the same resource, one with the longest remaining series of activities should be given priority. It is also worth to mention that this priority rule was first used by Brooks (Brooks, White 1965).

- b) LFT is the abbreviation of "Latest Finish Time". The latest finish time for each project is calculated, and the activity which has the smallest value of LFT is programmed first (Ahuja 1976).
- c) MNSLCK is the slack amount of activities. This term can also be defined as the importance level of processes. Slack is the difference in the early start time and latest start time of the activity. The activity, which has the least slack, is programmed first. David and Patterson (1975) used this priority rule in their studies to find the optimum project time:

$$S_{ij} = LS_{ij} - ES_{ij}. \quad (2)$$

1.2. Projects description

240 manufacturing or construction projects are programmed with various limited-resource conditions using three priority rules. The project characterizations are summarized in Table 1 as a sample and one project is shown in appendix. Each project is programmed under 2, 4, 6, 8 constraint. Daily resources constraints are given in Table 2.

2. Artificial neural network

The mathematical models named as artificial neural networks (ANN) were originally inspired by the organizational and functional structure of biological neurons. The network is then given a task whose nature is generally associated with the presence of a number of artificial neural network features. ANNs are automatically able to approximate the functional form that most closely characterizes the data (Rumelhart, McClelland 1986; Wasserman 1989). ANN architectures consist of three layers or more, which constitute of an input layer, an output layer and a number of hidden layers. The last group performs the task of connecting the neurons to each other via interconnections with modifiable weights (Fig. 1). In this context, the ANN architecture is frequently regarded as a fully interconnected feed forward multilayer perception. Furthermore, the ANN structure also exhibits a bias that is only connected to the neurons with modifiable weighted connections in the hidden and output layers. The exact

Table 1. Sample project (10 project) characterizations

Number of project	Number of activity	Number of joint	Project widths	CPM time (day)	Number of resource constrained	Resource constrained project time (day)		
						MRPL	LFT	MNSLCK
1	10	9	2	33	2	52	54	52
					4	43	43	43
					6	47	49	52
					8	50	50	50
2	15	13	3	47	2	60	61	62
					4	67	67	75
					6	74	74	74
					8	69	69	86
3	20	17	3	35	2	53	53	59
					4	55	52	57
					6	50	49	52
					8	57	61	58
4	25	22	4	39	2	51	51	55
					4	69	67	78
					6	75	81	83
					8	60	62	68
5	30	26	4	64	2	86	87	90
					4	96	91	104
					6	79	80	87
					8	109	113	121
6	35	31	4	68	2	98	88	101
					4	95	94	104
					6	101	100	116
					8	90	89	99
7	40	34	4	68	2	86	88	99
					4	114	107	118
					6	107	108	119
					8	104	98	106
8	45	38	4	80	2	107	105	107
					4	131	122	138
					6	126	127	143
					8	136	139	153
9	50	40	6	61	2	118	118	131
					4	119	127	141
					6	106	109	122
					8	123	120	134
10	55	46	6	71	2	116	118	130
					4	141	145	178
					6	117	118	143
					8	161	150	168

Table 2. Daily resource constraint (C) number

	Daily resource constraint number							
	C1	C2	C3	C4	C5	C6	C7	C8
Two resource constraint	5	3						
Four resource constraint	5	3	4	4				
Six resource constraint	5	3	4	4	5	4		
Eight resource constraint	5	3	4	4	5	4	6	5

number of neurons in each of the layers varies depending on the particular nature of the problem to be solved.

The most widely used training algorithm for multi-layered feed-forward networks is the back-propagation (BP) algorithm. The BP algorithm involves two phases. The first one is the forward phase where the activations are propagated from the input towards the output layer. The second one is the backward phase where the error between the observed actual value and the desired nominal value in the output layer are propagated backwards in order to modify the weights and the bias values. The inputs and the outputs of training and testing sets must be initialized before then the training a feed work network. In the forward phase, the weighted sum of input components is calculated as:

$$net_j = \sum_{i=1}^n w_{ij}x_i + bias_j, \quad (3)$$

where: net_j is the weighted sum of the j^{th} neuron for the input received from the preceding layer with n neurons; w_{ij} is the weight between the j^{th} neuron and the i^{th} neuron in the preceding layer; x_i is the output of the i^{th} neuron in the preceding layer. The output of the j^{th} neuron out_j is calculated with a sigmoid function and where γ is constant used to control the slope of the semi-linear region. $\psi(\gamma'w_i)$ is the sigmoid function defined as:

$$\psi(\gamma'w_i) = \frac{1}{1 + e^{-\gamma'w_i}}. \quad (4)$$

The training of the network is achieved by adjusting the weights and carried out through a large number of training sets and training cycles. The goal of the training procedure is to find the optimal set of weights, which would produce the right output for any input in the ideal case. Training the weights of the network is iteratively adjusted to capture the relationship between the input and output patterns (Moller 1993).

2.1. Computational experiments

The success project numbers of priority rules obtained from the programming are presented in Table 3.

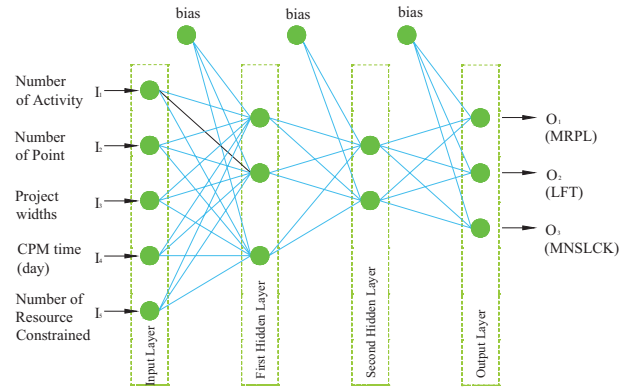


Fig. 1. General architecture of ANN models

Table 3. Success project number of priority rules

	MRPL	LFT	MNSLCK
The minimum project time	148	132	40
The minimum project time alone	92	76	12

The priority rules MRPL, LFT and MNSLCK yielded the shortened project duration for 148, 132 and 40 projects, respectively. A thorough reading of the table above demonstrates that multiple successful rules exist for some particular cases. Moreover, Table 3 also displays the number of projects that each priority rule delivered the shortest duration on a stand-alone basis. In this context, MRPL, LFT and MNSLCK yielded the shortened project duration in 92, 76 and 12 projects uniquely.

Although many artificial intelligent methods were used in the solving stage of project programming in the literature, there appeared no study in which guessing performance of priority rules are tested according to the general knowledge of the projects whose sources are limited. In the study, the ANN based model was applied to determine priority rules respect to performance of minimum project time. The data are divided into two parts as the training and testing sets. 202 projects are selected randomly as training set and employed to train NN based model. Testing set is constituted of 38 projects to test the NN based models. These projects, which are not used in the training process, are used to validate the generalization capability of NN based model. In the ANN modeling process, the input and output data sets for each parameter are normalized in following formulation:

$$D_t = 0.1 + 0.8 \frac{(D - D_{min})}{(D_{max} - D_{min})}, \quad (5)$$

where: D_t is the normalized value of a certain parameter; D is the measured value for this parameter; D_{min} and D_{max} are the minimum and maximum values in the database for this parameter, respectively.

The inputs of ANN models consist of four data defining the project characterization. The input parameters are number of activity, number of joints, CPM time and resource constrained (Table 1). The learning algorithm is used Scaled Conjugate Gradient (SCG). Each ANN model is initialized with different random weights.

3. Result and discussion

The ANN model was analysed with 1, 2, 3 and 4 layers and number of epochs at 1000, 2000 or 3000. Primarily, the ANN model’s training performance is investigated and the obtained results are given in Table 4. MSE is defined mean square error (6).

In view of the training MSE for the ANN model, it is seen that the model with 3 hidden layers and 2000 epochs displayed the best training performance. Figure 2 depicts the best ANN model’s training performance with respect to project time.

The evaluation and comparison of the prediction performance necessitates the introduction of an appropriate evaluation criterion. There is no consensus on the most appropriate measure to assess the performance of a forecasting technique. In this context, three different consistency measures are employed in the literature to compare the predictive power of ANN models with each other. In order, these criteria are the mean square error (MSE), the mean absolute error (MAE), and the mean absolute percentage error (MAPE). Even though these stated criteria are reliable measures of the prediction errors relative to the actual values, they are incapable of reflecting a model’s ability to forecast inflection points. The aforementioned values are obtained by the following procedure (Moller 1993):

The first measurement is the mean square error:

$$MSE = \sum_{t=1}^T \frac{(P_t - Z_t)^2}{T} \quad (6)$$

The second criterion is the mean absolute error:

$$MAE = \sum_{t=1}^T \frac{|P_t - Z_t|}{T} \quad (7)$$

The third criterion is the mean absolute percentage error:

$$MAPE = \frac{1}{T} \sum_{t=1}^T \frac{P_t - Z_t}{Z_t} \quad (8)$$

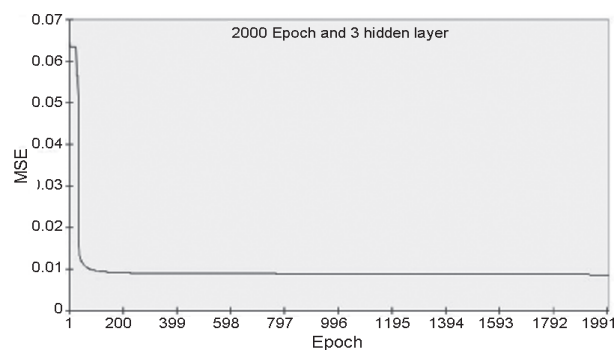


Fig. 2. ANN Model’s training performance

Table 4. Training performance of ANN models

Hidden number	Epoch number	Minimum MSE	Final MSE
4	3000	0.008621	0.008621
3	3000	0.008594	0.008594
2	3000	0.008577	0.008577
1	3000	0.008641	0.008641
4	2000	0.008650	0.008650
3	2000	0.008408	0.008408
2	2000	0.008629	0.008629
1	2000	0.008636	0.008636
4	1000	0.008821	0.008821
3	1000	0.008507	0.008507
2	1000	0.008507	0.008507
1	1000	0.008750	0.008750

where: P_t is the predicted value at time t ; Z_t is the actual value at time t ; and T is the number of predictions. The forecasting results of ANN models are summarized in Table 5.

According to project programming results, the priority rule named MRPL, which minimized the project duration in most projects, displayed a prediction performance of 0.702, surpassing LFT and MNSLCK priority rules, whose corresponding performance figures were 0.699 and 0.672, respectively. The forecasting performance is better when the values of MSE, NMSE and MAE are smaller. However, if the results are not consistent among these three, the MAE (Makridakis 1993) is chosen to be the benchmark with relative stable then other criteria.

Further research on the issue has revealed that the effectiveness of the priority rules used in heuristic methods highly depends on the project size and the number of available resources. Abbasi and Arabiat (2000) tested 60 projects in which the cash flow was defined by a priority rule that can be described as a combination of the LST and SPT methods. David and Patterson (1975) examined the success ratio of various priority rules in 83 different projects. According to the findings of this study, the MNSLCK and LFT priorities gave the best result in 24 and 12 projects, respectively. Ozdamar and Ulusoy made a similar analysis on 78 projects through parallel methods according to four different priority rules (Ozdamar, Ulusoy 1996). MNSLCK gave the best result in 24 projects and LFT in 22 projects. Schirmer (1999) scheduled his projects with 30 activities and used eight priority rules (Schirmer 1999). Out of these eight priority rules, the LFT priority yielded the highest level of performance in 42 projects whereas the MNSLCK priority gave the best result in 24 projects and the MTS priority in 23 projects. Klein (2000) used various priority rules for his study and found out that the LFT priority gave the

best performance in 36 projects, the MTS Priority in 20 projects, and the GRPW and LST priorities in 19 projects. When the parallel methods were used, the priority which gave the best performance was the LFT priority with 28 projects (Klein 2000). Kanit *et al.* (2009a) used 5 distinct priority rules in their studies which investigated the effects of project size and resource containments on project durations; as a result of which they reached the conclusion that MRPL priority rule exhibits the highest performance. A thorough literature review reveals

that there is no common, best priority rule in limited resource programming problems. It is clearly seen that the effectiveness of the priority rules differs from project to project (Kanit *et al.* 2009b).

The estimated measured values of project time using MRPL, LFT and MNSLCK priority rules are displayed Figures 3, 4 and 5, respectively. The performance of each priority rule is determined by the ratio of exceeding CPM duration. For this reason, MRPC/CPM, LFT/CPM, and MNSLCK/CPM ratios are given below.

Table 5. Comparison of observed and predicted data from ANN Models

	Epoch N.	Hidden N.	MSE	NMSE	MAE	Min Abs Error	Max Abs Error	r ²
MRPL	3000	4	0.121794	0.521358	0.272362	0.004296	0.751947	0.697507
LFT	3000	4	0.146849	0.591857	0.318848	0.050827	0.732619	0.659965
MNSLCK	3000	4	0.144307	0.637174	0.325134	0.001065	0.691800	0.638021
MRPL	3000	3	0.123853	0.530170	0.273185	0.003511	0.736846	0.692529
LFT	3000	3	0.148864	0.599982	0.320075	0.045757	0.739981	0.654734
MNSLCK	3000	3	0.145414	0.642061	0.325471	0.002573	0.697443	0.635984
MRPL	3000	2	0.123926	0.530486	0.273713	0.003547	0.736044	0.692395
LFT	3000	2	0.148057	0.596728	0.319462	0.053261	0.738821	0.657188
MNSLCK	3000	2	0.144778	0.639256	0.325171	0.001937	0.696526	0.637527
MRPL	3000	1	0.123653	0.529318	0.274021	0.005243	0.737154	0.692315
LFT	3000	1	0.148390	0.598071	0.320335	0.050806	0.739950	0.655655
MNSLCK	3000	1	0.145593	0.642851	0.325898	0.003725	0.697203	0.634832
MRPL	2000	4	0.122196	0.523080	0.273402	0.004968	0.735611	0.696143
LFT	2000	4	0.147940	0.596257	0.319984	0.050251	0.738520	0.656782
MNSLCK	2000	4	0.144752	0.639137	0.325462	0.001536	0.696002	0.636497
MRPL	2000	3	0.120053	0.513906	0.270886	0.001162	0.644464	0.702533
LFT	2000	3	0.146818	0.591733	0.315507	0.045159	0.646787	0.669854
MNSLCK	2000	3	0.142286	0.628252	0.324354	0.002689	0.605074	0.672665
MRPL	2000	2	0.124277	0.531987	0.274151	0.005075	0.740410	0.690675
LFT	2000	2	0.148885	0.600067	0.320226	0.046574	0.742038	0.654456
MNSLCK	2000	2	0.145876	0.644101	0.325649	0.005365	0.700662	0.634294
MRPL	2000	1	0.123260	0.527632	0.274043	0.006475	0.737548	0.693366
LFT	2000	1	0.148589	0.598871	0.320686	0.052830	0.742011	0.655123
MNSLCK	2000	1	0.145412	0.642053	0.325937	0.003248	0.698800	0.635097
MRPL	1000	4	0.120698	0.516665	0.273966	0.008910	0.745334	0.699709
LFT	1000	4	0.147605	0.594907	0.321226	0.041748	0.746409	0.656973
MNSLCK	1000	4	0.143457	0.633419	0.32645	0.007573	0.704818	0.637785
MRPL	1000	3	0.120622	0.51634	0.274215	0.006217	0.739294	0.700197
LFT	1000	3	0.147486	0.594426	0.321255	0.043433	0.741609	0.657452
MNSLCK	1000	3	0.143608	0.634089	0.327338	0.002395	0.699658	0.637487
MRPL	1000	2	0.122011	0.522288	0.271685	0.004401	0.731889	0.697805
LFT	1000	2	0.146232	0.589372	0.317237	0.043676	0.732689	0.662315
MNSLCK	1000	2	0.143298	0.632718	0.324983	0.004474	0.687598	0.642805
MRPL	1000	1	0.121787	0.521328	0.274102	0.003358	0.741996	0.696943
LFT	1000	1	0.147687	0.595237	0.321220	0.045633	0.74253	0.657059
MNSLCK	1000	1	0.143297	0.632714	0.326687	0.008026	0.701876	0.638224

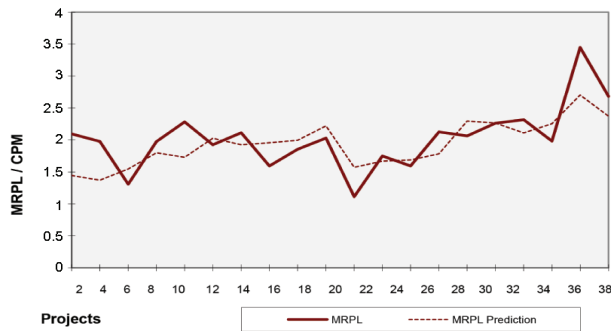


Fig. 3. The estimated values for project time as per the MRPL priority rule in testing set

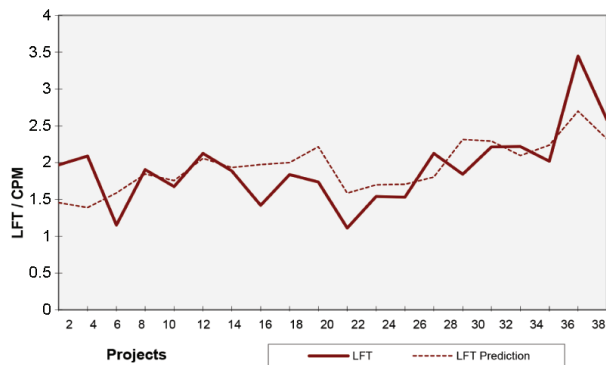


Fig. 4. The estimated values for project time as per the LFT priority rule in testing set



Fig. 5. The estimated values for project time as per the MNSLCK priority rule in testing set

Conclusions

It is quite difficult to estimate which project time and priority rule exhibits the best performance under the conditions of constrained resources. An extensive review of the similar studies in the literature reveals that the performance of priority rules depend on several factors such as project type, the scope of project, number of resources and restrictions etc. The conclusion drawn by the researchers underscores that under the assumption of constrained resources, it is almost impossible to attribute high performance to any particular priority rule. In the study, 240 projects with different types, sizes and objec-

tives of usage were analysed in accordance with three distinct priority rules (MRPL, LFT and MNSLCK) under the condition of constrained resources. In the next stage, an ANN model that utilized the project characteristics to estimate the project time for each priority rule was developed. The model with 3 hidden layers and 2000 epochs was found to have highest performance among the training and forecasting models. In this particular model, MRPL displayed an correlation coefficient of 0.702, while the corresponding values for LFT and MNSLCK priority rules were measured as 0.669 and 0.672, respectively. In the constrained resource project programming problems where the minimum project time depends on a wide range of factors, an estimation performance on the order of 0.70 is deemed as a successful level.

After the introduction of the estimation model based on ANN, the project time of resource constrained project scheduling problems will be known prior to advancing to the programming stage. Consequently, it will be possible to determine which particular priority rule will minimize the project time and also estimate the level of resource constraints before proceeding with the programming stage. This estimation technique, in turn, will help to prevent unnecessary losses of time and money in industries using resource constrained project programming, especially in the construction sector.

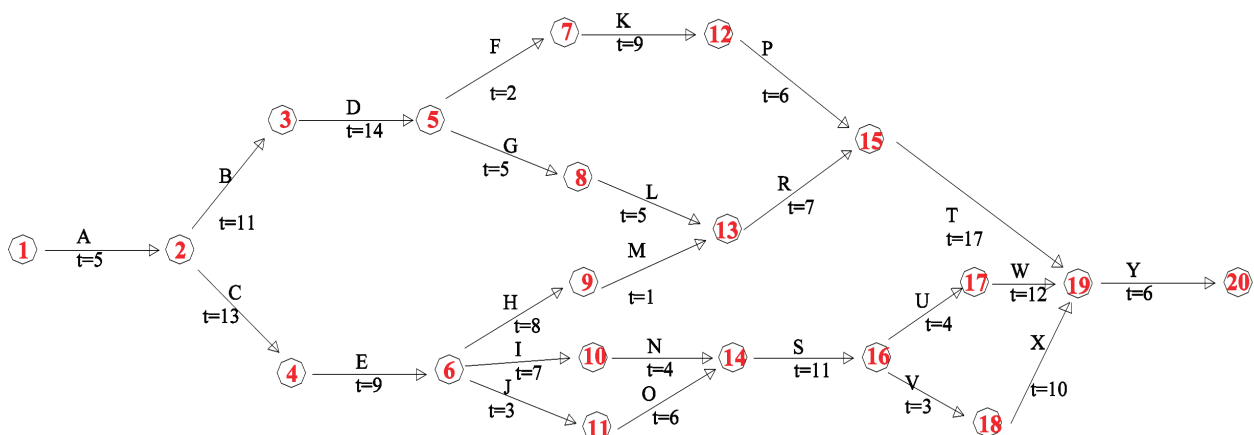
References

- Abbasi, G. Y.; Arabiat, Y. A. 2000. A heuristic to maximize the net present value for resource-constrained project scheduling problems, *International Journal of Project Management* 32(2): 17–24.
- Ahuja, H. N. 1976. *Construction performance control by network*. New York: Wiley.
- Brooks, G. H.; White, C. R. 1965. An algorithm for finding optimal or near optimal solutions to the production scheduling problem, *Journal of Industrial Engineering* 16(1): 34–40.
- Brucker, P.; Drexel, A. 1999. Resource-constrained project scheduling: notation, classification, models, and methods, *European Journal of Operational Research* 112(1): 3–41. [http://dx.doi.org/10.1016/S0377-2217\(98\)00204-5](http://dx.doi.org/10.1016/S0377-2217(98)00204-5)
- Bruni, M. E.; Beraldi, P.; Guerriero, F.; Pinto, E. 2011. A heuristic approach for resource constrained project scheduling with uncertain activity durations, *Computers & Operations Research* 38(9): 1305–1318. <http://dx.doi.org/10.1016/j.cor.2010.12.004>
- Chassiakos, A. P.; Sakellariopoulos, S. P. 2005. Time-cost optimization of construction projects with generalized activity constraints, *Journal of Construction Engineering and Management* 131(10): 365–380. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2005\)131:10\(1115\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2005)131:10(1115))
- David, E. W.; Patterson, J. H. 1975. A comparison of heuristic and optimum solutions in resource-constrained project scheduling, *Management Science* 21(8): 944–955. <http://dx.doi.org/10.1287/mnsc.21.8.944>
- Demeulemeester, E.; Herroelen, W. A. 1992. Branch and bound procedure for the multiple resource-constrained project scheduling problems, *Management Science* 38(12): 1790–1803. <http://dx.doi.org/10.1287/mnsc.38.12.1803>
- Idoro, G. I. 2012. Comparing the planning and performance of direct labour and design-bid-build construction projects in Nigeria, *Journal of Civil Engineering and Management* 18(2): 184–196. <http://dx.doi.org/10.3846/13923730.2012.671283>

- Jaśkowski, P.; Sobotka, A. 2004. Scheduling construction projects with resources accessibility limited and changeable in time, *Journal of Civil Engineering and Management* 10(4): 267–276.
<http://dx.doi.org/10.1080/13923730.2004.9636319>
- Kanıt, R.; Gündüz, M.; Özkan, Ö. 2009a. Investigating the effectiveness of certain priority rules on resource scheduling of housing estate projects, *Journal of Construction Engineering and Management* ASCE 135(7): 609–613.
[http://dx.doi.org/10.1061/\(ASCE\)CO.1943-7862.0000021](http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000021)
- Kanıt, R.; Özkan, Ö.; Gündüz, M. 2009b. Effect of project size and resource constraints on project duration through priority rule-based heuristics, *AI Review Journal* 32(1–4): 115–123.
- Klein, R. 2000. Project scheduling with time-varying resource constraints, *International Journal of Production Research* 38(16): 3937–3952.
<http://dx.doi.org/10.1080/00207540050176094>
- Kolish, R.; Hartman, S. 1998. *Heuristic algorithm for solving resource constrained project scheduling problem*. Technical Paper, No 469. Kiel University.
- Leu, S. S.; Chen, A. T.; Yang, C. H. 1999. Fuzzy optimal model for resource-constrained construction scheduling, *Journal of Computing in Civil Engineering* 13(3): 207–216.
[http://dx.doi.org/10.1061/\(ASCE\)0887-3801\(1999\)13:3\(207\)](http://dx.doi.org/10.1061/(ASCE)0887-3801(1999)13:3(207))
- Makridakis, S. 1993. Accuracy measures: theoretical and practical concerns, *International Journal of Forecasting* 9(4): 527–529.
[http://dx.doi.org/10.1016/0169-2070\(93\)90079-3](http://dx.doi.org/10.1016/0169-2070(93)90079-3)
- Moller, A. F. 1993. A scaled conjugate gradient algorithm for fast supervised learning, *Neural Networks* 6(4): 525–533.
[http://dx.doi.org/10.1016/S0893-6080\(05\)80056-5](http://dx.doi.org/10.1016/S0893-6080(05)80056-5)
- Nassar, K. M.; Hegab, M. Y. 2006. Developing a complexity measure for project schedules, *Journal of Construction Engineering and Management* 132(6): 551–564.
[http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2006\)132:6\(554\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2006)132:6(554))
- Ozdamar, L.; Ulusoy, G. 1996a. An iterative local constraint based analysis for solving the resource constrained project scheduling problem, *Journal of Operations Management* 14(3): 193–208. [http://dx.doi.org/10.1016/0272-6963\(95\)00015-1](http://dx.doi.org/10.1016/0272-6963(95)00015-1)
- Ozdamar, L.; Ulusoy, G. 1996b. A note on an iterative forward/backward scheduling technique with reference to a procedure by Li and Willis, *European Journal of Operational Research* 89(2): 400–407.
[http://dx.doi.org/10.1016/0377-2217\(94\)00272-X](http://dx.doi.org/10.1016/0377-2217(94)00272-X)
- Padman, R.; Smith-Daniels, D. E. 1993. Early-tardy cost tradeoffs in resource constrained projects with cash flows: an optimization-guided heuristic approach, *European Journal of Operational Research* 64(2): 295–311.
[http://dx.doi.org/10.1016/0377-2217\(93\)90184-O](http://dx.doi.org/10.1016/0377-2217(93)90184-O)
- Rogalska, M.; Hejducki, Z. 2007. Time buffers in construction process scheduling, *Journal of Civil Engineering and Management* 8(2): 143–148.
- Rumelhart, D.; McClelland, J. 1986. *Parallel distributed processing*. Cambridge: MIT Press.
- Schirmer, A. 1999. Resource constrained project scheduling: an evaluation of adaptive control schemes for parameterized sampling heuristic, *International Journal of Operations & Production Management* 39(7): 1343–1365.
- Slowinski, R. 1980. Two approaches to problems of resource allocation among project activities a comparative study, *Journal of the Operational Research Society* 31(8): 711–723. <http://dx.doi.org/10.2307/2581688>
- Talbot, F. B. 1982. Resource constrained project scheduling with time-resource tradeoffs: the non-preemptive case, *Management Science* 28(10): 1197–1210.
<http://dx.doi.org/10.1287/mnsc.28.10.1197>
- Ulusoy, G.; Ozdamar, L. 1989. Heuristic performance and network resource characteristics in resource constrained project scheduling, *Journal of Operations Research Society* 40(12): 1145–1152. <http://dx.doi.org/10.2307/2582924>
- Wasserman, P. D. 1989. *Neural computing: theory and practice*. New York: Van Nostrand Reinhold. 230 p.
- Yang, B.; Geunes, J.; O'Brien, W. 2001. *Resource-constrained project scheduling: past work and new directions*, Research Report 2001-6, NSF Grant, CMS-0122193.
- Yang, K. K.; Talbot, F. B.; Patterson, J. H. 1993. Scheduling a project to maximize its net present value: an integer programming approach, *European Journal of Operational Research* 64(2): 188–198.
[http://dx.doi.org/10.1016/0377-2217\(93\)90176-N](http://dx.doi.org/10.1016/0377-2217(93)90176-N)

Appendix

Sample project



Ömer ÖZKAN. He is a Visiting Researcher in University of Pittsburgh. He is an Assoc. Prof. Dr in the Technology Faculty, Civil Engineering Department at Sakarya University, Turkey. His research interests include the construction management, particularly resources constrained project scheduling.

Ümit GÜLÇİÇEK. He is a PhD student and works as an instructor in Kahramanmaraş Sutcu Imam University. His research interests include the construction management, cost and forecasting of construction and particularly resources constrained project scheduling.