Resource Constrained Project Scheduling Using Mean Field Annealing Neural Networks

Mohammad Jaberi
Department of Computers, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran
mohammadjaberi_61@yahoo.com

***Abstract*— A Potts mean field feedback artificial neural network algorithm is developed and explored for the resource constrained project scheduling problem. A convenient encoding of inequality constraints is achieved by means of multilinear penalty function. An approximate energy minimum is obtained by iterating a set of Potts means field equation, is combination with annealing. Priority rule-based heuristics are the most widely used scheduling methods though their performance depends on the characteristics of the projects. To overcome this deficiency, a Potts mean field feedback artificial neural network is designed and integrated into the scheduling scheme so as to automatically select the suitable activity for each stage of project scheduling. Testing on Paterson's classic test problems and comparison with other exact method how that the proposed Potts mean field annealing neural network based heuristic is able to improve the performance of project scheduling.**

***Keywords*—** Neural Network, Mean Field Theory, Potts Mean Field Theory, Resource-constrained Project Scheduling and Priority Rule-Based Heuristic

I. INTRODUCTION

The resource-constrained project scheduling problem (RCPSP) is a typical scheduling problem which allocates scarce resource over time to perform a set of activities in order to minimize the project duration. The problem is known to be strongly NPhard. Many optimization methods have been proposed for the problem such as exact, priority rule based heuristic and metaheuristic methods have been proposed for the resource-constrained project scheduling problem (RCPSP).

The exact methods often adopt mathematical models such as integer programming, Talbot (1982), and dynamic programming, Gavish and Pirkul (1991), or are based on implic enumeration with branch and bound by considering the RCPSP as NP-hard problem. But the exact methods may be computationally infeasible or face combinatorial explosion problem if the practical projects under study are larger or more complicated, Leu and Yang (1999), Chan ,Chua and Kannan (1996).

The general heuristic methods adopt priority rules reflecting one or multiple factors such as activity's critical index, duration, and minimum late finish time in generation of schedules, such as the ones used by Boctor (1990), Padilla and Carr (1991), Bell and Han (1991) and Sampson and Weiss (1993)). However, there is little basis for choosing one among different heuristic rules, and no priority rule dominates all other or performs consistently better than others, Davis and Patterson (1975). Moreover, the general heuristic methods may be trapped within local optima, Lee and Kim (1996).

The metaheuristic methods or the new generation of heuristic algorithms normally include simulated annealing (SA), tabu search (TS) and genetic algorithm (GA). SA searches for better solutions through repetitive improvement (or local alternation) on current solutions. Boctor (1990), Lee and Kim (1996) and Bouleimen and Lecocq (1998) have applied SA for the RCPSP. TS start with a feasible solution and keep improving it in the successive iterations so that a local optimum may be escaped in pursuit of a global optimum. Its application to the RCPSP includes the works of Pinson et al. (1994), Lee and Kim (1996) and Baar et al. (1998). GA is based on the mechanisms of evolution and natural genetics and has been applied to solve the RCPSP Lee and Kim (1996). The three metaheuristic methods have some common features such as starting with initial solutions and updating (or improvement) them from iteration to iteration. Comparisons of the solution-solving schemes for the RCPSP show that GA and SA have better performance than TS in addition that the metaheuristic methods generally outperform the exact or heuristic methods.

Feedback artificial neural network have turned out to be powerful in finding good approximentive solution to difficult combinatorial optimization problem, Hopfield and Tank (1985), Peterson and soderberg (1989), and Gislen, Peterson and Soderberg (1992). The idea it to map the problem onto binary or M-state neurons (spin variables) with an appropriate energy function. The system is relaxed using mean field theory (MFT) techniques in order to avoid local minima. This procedure, sometimes called mean field annealing (MFA) give an approximate global minimum of the energy. In This paper we extend integrate priority rule based heuristics into Potts-MFA to resource-constrained project scheduling.

II. PROBLEM DESCRIPTION (RCPSP FORMULATION)

The RCPSP is normally characterized by objective functions, features of resources, and preemption conditions, Lee and Kim (1996). Minimization of project duration is often used as an objective of the RCPSP, while other objectives such as minimization of total project cost and leveling of resource usage are also considered. Resources involved in a construction project can be renewable (i.e., recoverable after serving an activity, such as equipment or crew) or nonrenewable (i.e., limited in amount over project process and not recoverable, such as cement or sand). Preemption means the activities (e.g., frame installing) in progress can be interrupted, while non preemption means the activities (e.g., concreting) are not allowed to stop once in progress. The classical RCPSP that considers the renewable resources, non-preemption and minimizing of project duration can be formulated as follows, Talbot (1982):

$$Min\left\{Max\ F_j \,\middle|\, j = 1,2,......,N\right\} \quad (1)$$

Subject to:

$$F_k \le F_j - D_j, \forall j \in P_k; j = 1,2....,N \quad (2)$$

$$\sum\nolimits_{A_t} r_{ij} \le R_i \;\; i = 1,2.........K \;\; t = S_1, S_2, ...S_N \quad (3)$$

where N is the number of the activities involved in a project and F_j is the finish time of activity a_j, D_j is the duration of activity a_j, P_k is a set of preceding activities (or predecessors) of activity a_j, R_j is available amount of resource i, and i is the number of the resource types; r_{ij} is the amount of resource i required by activity a_j, and A_t is a set of ongoing activities at t, and $S_j (= F_j - D_j)$ is the start time of activity a_j. Formula (1) represents the objective, while formulas (2) and (3), respectively, represent precedence constraints and resource constraints.

Definition: $x_{j,t}$ is a variable that values 1 when j activity in completed in t time (0 otherwise). Then i resource quantity needed by j activity during the $[t-1, t]$ interval is:

$$\left(r_{ij} \sum_{u=t}^{t+D_j-1} t.x_{u,t} \right)$$

If H is a project duration upper bound, a simple but not very tight value may be obtained using next expression:

$$H = \sum_{j=1}^{n} D_j$$

Then time to complete j activity can be expressed as:

$$\sum_{t=1}^{H} t.x_{k,t}$$

And the minimizing of project duration or summation of time to complete all activity can be expressed as:

$$\sum_{t=1}^{H} t.x_{n+1,t} \text{ or } \sum_{t=1}^{H}\sum_{j=1}^{n+1} t.x_{j,t}$$

An Integer programming formulation can now be formulated as follows:

$$Min \sum_{t=1}^{H} t.x_{n+1,t} \quad (4)$$

$$\sum_{t=1}^{H} t.x_{j,t} + D_k - \sum_{t=1}^{H} t.x_{j,t} \le 0 \qquad \forall j \to k \in A \quad (5)$$

$$\sum_{j=1}^{n} \left(r_{i,j} . \sum_{u=t}^{t+D_j-1} t.x_{u,t} \right) \le R_i \qquad \forall i,t \quad (6)$$

$$\sum_{t=1}^{H} t.x_{j,t} = 1 \quad (7)$$

Schedule objective is to minimizing of project duration. Set (5) constraints assure that precedence restrictions are met, j activity is followed by k activity, Set (6) constraints ensures that the total i resource demand at t time, cannot exceed its resource availability. The third set of constraint (7) ensures that every activity is processed. When the number of activities is large and the planning horizon is long, the RCPSP is usually solved using heuristics and metaheuristics, which already proved to provide effective solutions.

III. PRIORITY RULE-BASED HEURISTICS

Priority rule-based heuristics consist of at least two components, including a schedule generation scheme (SGS) and priority rules. An SGS determines how a schedule is constructed gradually, building a feasible full schedule for all activities by augmenting a partial schedule covering only a subset of activities in a stage-wise manner. Two schemes are usually distinguished. In the serial SGS, a schedule is built by selecting the eligible activities in order and scheduling them one at a stage as soon as possible without violating the constraints. In the parallel SGS, a schedule proceeds by considering the time periods in chronological order and in each period all eligible activities are attempted to start at that time if resource availability allows. For each feasible RCPSP instance, a serial SGS searches among the set of active schedules which always contains at least one optimal schedule for project duration minimization, Kolisch (1996). Therefore, the serial SGS is adopted in this paper.

The serial SGS divides the set of activities into three disjoint subsets: scheduled, eligible, and ineligible. An activity that is already in the partial schedule is considered as scheduled. Otherwise, an activity is called eligible if all its predecessors are scheduled and ineligible otherwise. The subsets of eligible and ineligible activities form the subset of unscheduled activities. The scheme proceeds in $N = J$ stages, indexed by n. On the $n-th$ stage, the subset of scheduled activities is denoted as S_n and to the subset of eligible activities as decision set D_n. On each stage, if more than one activity is eligible, one activity j from D_n is selected using a priority rule and scheduled to begin at its earliest feasible start time. Then activity j is moved from D_n to S_n which may render some ineligible activities eligible if now all their predecessors are scheduled. The scheme terminates on stage N when all activities are scheduled.

Priority rules serve to resolve conflicts between activities competing for the allocation of scarce resources. In situations where the decision set contains more than one candidate, priority values are calculated from numerical measures which are related to properties of the activities, the complete project, or the incumbent partial schedule.

Some well-known priority rules are listed in Table 1, in which ES_j and LS_j denote the earliest start time and latest start time for activity j according to the critical path method (CPM). Latest start or finish time (LST, LFT), slack (SLK), Kolisch (1996).

TABLE 1
PRIORITY RULES FOR RCPSP HEURISTICS

Rule	Extremum	Definition
Latest start time(LST)	MIN	LS_j
Latest finish time(LFT)	MIN	$LS_j + p_j$
Minimal slack (SLK)	MIN	$LS_j + ES_j$

IV. OPTIMIZATION WITH MEAN FILED ANNEALING

Recurrent networks appear in the context of associative memories and difficult optimization problems, Hopfield and Tank, (1985). Simple models for magnetic systems (spin glasses) have a lot in common with recurrent networks-with an atomic spin seen as analogous to the firing state of a neuron – and have therefore been the source of much inspiration for neural network studies. The Hopfield model is based on the energy function:

$$E = -\frac{1}{2}\sum_{i \neq j} w_{ij} s_i s_j \tag{8}$$

In terms of binary variables (Ising neurons) $s_i = \pm 1$ (or 0,1) with symmetric weights w_{ij}. With an appropriate choice of weights depending on the stored patterns, the model serves as an associative memory, with an asynchronous dynamics that locally minimizes E, Peterson and soderberg (1989):

$$s_i(t+1) = \operatorname{sgn}\left(\sum_{j \neq i} w_{ij} s_i s_j\right) \tag{9}$$

V. OPTIMIZATION WITH ISING NEURAL NETWORKS

The archetype of an ANN for optimization is based on a Hopfield-type energy function, adapted to a specific problem by a dedicated choice of weights. With a slightly modified, softer, MF dynamics, with sgn (0) replaced by $\tanh(0/T)$, combined with annealing, the resulting MF neurons will relax to a stable configuration representing a tentative solution to the problem. The key problem here is to reach the global minimum or at least a very low-lying local minimum.

If one attempts to minimize E according to a local optimization rule, the system will very likely end up in some local minimum close starting point, which is not desired.

A better strategy is to employ a stochastic algorithm that allows for uphill moves. One such method is Simulated Annealing (SA), in which configurations are generated according to the Boltzmann distribution, Kirkpatrik, Gelatt, and Vecchi (1983):

$$P[s] = \frac{1}{z} e^{-E[s]/T} \tag{10}$$

With neighborhood search methods. In Equation (10), Z is the partition function

$$Z = \sum_{[s]} e^{-E[s]}/T \tag{11}$$

And T is a temperature representing the noise level of system. $T = 0$ The Boltzmann distributed becomes concentrated to the configuration minimizing E. If configuration are generated with a slowly decreasing T (annealing), they are less likely to get stuck in local minima than if T is set to 0 from the start. Needless to say, such a procedure can be very CPU consuming.

The MF approach aims at approximating the stochastic SA method with a set of deterministic equations. To this end, introduce for each spin s_j a new variable v_i, living in a linear space containing the compact state-space (± 1) of the spin, and set it equal to the spin with a direct delta function. Then Z takes the form

$$Z = \sum_{[s]} \int d[v] e^{-E[v]/T} \prod_i \delta(s_i - v_i) \tag{12}$$

Fourier expending the delta functions in terms of conjugate variables u_i gives

$$Z \propto \sum_{[s]} \int d[u] \int d[u] e^{-E[v]/T} \prod_i e^{u_i(s_i - v_i)} \tag{13}$$

Then carry out the sum over $[s]$, and write the product as a sum in the exponent:

$$Z \propto \int d[v] \int d[u] e^{-E[v]/T - \sum_i u_i v_i + \sum_i \log\cosh u_i} \quad (14)$$

The original petition function is now rewritten entirely in terms of the new variables $[u, v]$, with an effective energy in the exponent. So far no approximation has been made.

We next assume that Z in Equation (14) is dominated by external value of the integrand, occurring for v_i satisfying the meaning the mean-field equations

$$v_i = thanh\left(-\frac{\partial E[v]}{\partial v_i} \Big/ T\right) \equiv \tanh\left(\frac{1}{T}\sum_j w_{ij} v_j\right) \quad (15)$$

The resulting MF variable v_i can be seen as approximations to the thermal averages $\langle s_i \rangle_T$ of the original binary spin.

The MF equations (15) are solved iteratively, either synchronously or asynchronously, under annealing in T. This yields a deterministic dynamics, characteristic of a recurrent ANN. High temperatures correspond to very smooth sigmoid $\tanh(0/T)$, while in the low-temperature limit the step function of Equation (9) is recovered, Bahreininejad and Topping (1997).

VI. OPTIMIZATION WITH POTTS NEURAL NETWORK

For many optimization problems, an encoding in terms of binary elementary variables in natural. However, there are many problems where the natural elementary decisions are of the type one-of-$k\rangle 2$.

Early attempts to approach such problems by neural network methods used neuron multiplexing, where for each elementary k-fold decision, a set of k binary 0/1- neurons was used, with the additional constraint that precisely one of them be on (=1). These syntax constraints were implemented in a soft way as penalty terms. this approach dose not yields high-quality solutions in a parameter-robust way.

An alternative encoding is to use Potts neurons with the syntax constraint built in. In this way the dynamics is confined to relevant parts of the solution space (Figure1), leading to dramatically improved performance Peterson and soderberg (1989).

VII. POTTS SPINS

A k-state Potts spin is variable has k possible values (states). For our purposes, the best representation is in terms of a vector in the Euclidean space ε_k. Thus, denoting a spin variable by $s = (s_1, s_2 ....s_k)$ the a the principal unit vector, defined by $s_a = 0, s_b = 1$ for $a \neq b$ these vectors points to the corners of a regular k-simplex (see Figure 1 the case of $k = 3$). They are all normalized and mutually orthogonal, and fulfill in addition the syntax $\sum_a s_a = 1$.

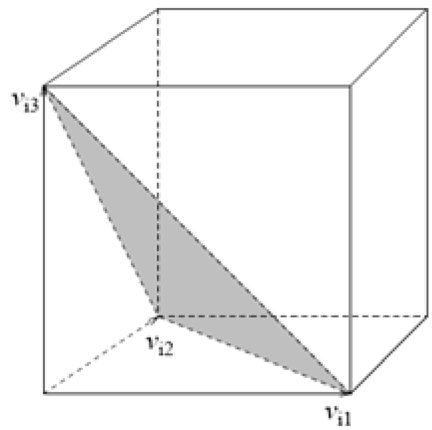

Fig. 1. The volume of solutions corresponding to the neuron multiplexing encoding for k=3. The shaded plane corresponds to the solution space of corresponding Potts encoding

The MF equations for a system of Potts spins s_i with a given energy function $E(s)$ are derived following the same path as in the Ising neuron case Rewrite the partition function as an integral over u_i and v_i approximate with the maximum value of the integrand.

$$u_{ia} = -\frac{\partial E_G}{\partial v_{ia}} \Big/ T \quad (16)$$

$$v_{ia} = \frac{e^{u_{ia}}}{\sum_b e^{u_{ia}}} \quad (17)$$

Form which it follows that the Potts neurons v_i whish approximate the thermal average of s_i, satisfy

$$v_{ia} > 0, \qquad \sum_a v_{ia} = 1 \quad (18)$$

One can think of the neuron component v_{ia} as the probability for the i the Potts spin to be in state a. for $k = 2$ one recovers the formulation of the Ising case in a slightly disguised form.

VIII. REFINEMENTS AND GENERALIZATIONS

In this section, we will discuss possible complications that arise in optimization applications and require special care in one way or another.

A. *None-Quadratic Energy Functions*

Not all optimization problems can be encoded in terms of a quadratic energy function, even though the state-space can be encoded in terms of set Potts neurons. This presents no principal difficulty, and one can still use Equation 8. However, a possible practical problem arises from the induced self-coupling in the energy function (terms with nonlinearities in a single spin), that might affect performance. With a quadratic E self-couplings can be avoided by removing all diagonal terms, $s_{ia}s_{ib} \to \delta_{ab}s_{ia}$. Such a procedure can be generalized to any polynomial E. Although polynomial of at most degree n, this can be difficult in practice for large N Peterson and soderberg (1989).

An efficient and general method for avoiding self-couplings altogether is to replace the derivative in Equation (16) by a difference:

$$u_{ia} = -\frac{1}{T}\left[\left(E_G \right)_{|v_{ia}=1} - \left(E_G \right)_{|v_{ia}=0} \right] \tag{19}$$

B. Inequality Constraints

In the problem mentioned in the previous sections, the constraints considered were all the equality Type, $f(s) = 0$, that could be implemented with quadratic penalty terms $\alpha\, f(s)^2$. However, in many optimization problems, in particular those of resource allocation type one has to deal with inequalities. An inequality constraint, $g(s) \le 0$ can be implemented with penalty term e.g. proportional to

$$\phi(x) = f(x)\theta(x) \tag{20}$$

$$f(x) = x + \frac{1}{2}x^2 \tag{21}$$

With θ the Heaviside step function: $\theta(x) = 1$ if $x \rangle 0$ and 0 otherwise. Of course, such a non-polynomial term in the energy function must be handled using Eq. (19).

IX. POTTS-MEAN FIELD APPROACH TO RCPSP

Resource-Constrained Project Scheduling Problems are NP-Complete and one is stuck to approximate solutions for large N problems. We will use the Potts Mean Field Theory equations to construct a polynomial-time algorithm. Resource-Constrained Project Scheduling Problem is mapped onto generic energy function E_G with M-state Potts neurons. E_G Defined by Eq. (14):

$$E_G = \frac{1}{2}\sum_{j=1}^{n+1}\sum_{t=1}^{H} w_{j,t}.s_{j,t} + \frac{\alpha}{2}\sum_{t=1}^{H}\left(\sum_{j=1}^{n+1} s_{j,t}\right)^2 + \beta\left(\sum_{k=1}^{H}\sum_{j\in predk}\phi\left(\sum_{t=1}^{H} t.s_{j,t} + D_k - \sum_{t=1}^{H} t.s_{k,t}\right)\right) + \gamma\left(\sum_{i=1}^{M}\sum_{t=1}^{H}\phi\left(\left(\sum_{j=1}^{n+1} r_{ij}\left(\sum_{u=t}^{t+D_k-1} s_{j,u}\right)\right) - R_i\right)\right) \tag{22}$$

The derivative $\frac{\partial E_G}{\partial v_{ia}}$ is treated exactly as the in the RCPSP case. Self-coupling terms are avoided by a linear approximation of $E_G(v_i)$, and one obtains

$$-\frac{\partial E_G}{\partial v_{ia}} = w_{j,k} + \alpha s_{j,k} + \beta\left(\sum_{j\in predk}\phi\left(\sum_{t=1}^{H} t.s_{j,k} + D_k - \sum_{t=1}^{H} t.s_{k,t}\right)_{|x_{j,t}=1} - \phi\left(\sum_{t=1}^{H} t.s_{j,k} + D_k - \sum_{t=1}^{H} t.s_{k,t}\right)_{|x_{j,t}=0}\right) + \left(\left(\sum_{i=1}^{M}\sum_{t=1}^{H}\phi\left(\left(\sum_{j=1}^{n+1} r_{j,k}\left(\sum_{u=t}^{t+D_k-1} s_{j,u}\right)\right) - R_i\right)_{|x_{j,t}=1}\right) - \left(\sum_{i=1}^{M}\sum_{t=1}^{H}\phi\left(\left(\sum_{j=1}^{n+1} r_{j,k}\left(\sum_{u=t}^{t+D_k-1} s_{j,u}\right)\right) - R_i\right)_{|x_{j,t}=0}\right)\right) \tag{23}$$

Thus, Table 2 shown as integrated priority rule based heuristics into Potts-MFA to resource-constrained project scheduling.

TABLE 2
POTTS-MFA FOR RESOURCE-CONSTRAINED PROJECT SCHEDULING

-Choose problem (Find the energy Function)
-Set up the weight matrix T
-Initialize the parameters of problem such as T, α, β, γ
  -For $n-2$ to j
    Select a priority rule based heuristic
    Calculate D_n
    Select j from D_n according to the priority rule selected
    s_n - s_n u { j }
  -end for
-Initialize the neurons (v_{ia}) with $\frac{1}{n+1}+$ plus a small noise factor increment
-Begin the annealing function
-Until sigma = $\frac{1}{n+1}\sum_{j=1}^{n+1}\sum_{t=1}^{n+1} v_{j,t}^2 \ge 0.99$ do (sigma=Saturation Criteria)
  - While Delta >0.01 do
    Cost=0
    Calculate equalities (16, 17)
    Cost= $u_{j,t}$
    Delta= $|\cos t - old \cos t|$
    Cost=Old Cost
  - $T_{k+1} = 0.98T_k$
  - $\alpha = Max(\alpha * 1.5, 1)$
  - $\beta = Max(\beta * 1.5, 1)$
- $\gamma = Max(\gamma * 1.5, 1)$

X. TESTING AND COMPARISON

In this section we present the result of the computational tests and comparisons with the best published algorithms. The approach is coded in Visual C# and run on an Intel Core 2 Dou CPU T7500 2.2 GHz personal computer.

The Paterson's test problems are adopted to test the proposed heuristic, see Patterson (1984). The problem set includes 110 scheduling problems, each with 3-50 activities and 1-4 renewable resources. The proposed heuristic is applied to solve each test problem and the priority rule-based heuristics listed in Table 1 are also applied for comparison. The computation results are shown in Table 3. The 74.5% of schedules generated by the proposed Potts-MFA are optimal. And the average error to optimal solutions is 1.1%, which is much smaller than the errors of other tested heuristics. The descriptive statistics verified sufficiently that the method proposed in this paper is able to improve the performance of resource constrained project scheduling. It is also noticed that for some instances the proposed ANN-based heuristic failed to generate near optimal schedules. The maximum error is 18.2%, for which the test problem has optimal project duration of 11 time units and the proposed heuristic generates a schedule of 13. Besides, 23.6% schedules have errors within 10.0%, which indicates the necessity of further improvement.

TABLE 3
DESCRIPTIVE STATISTICS OF RESULTS OF HEURISTICS

Heuristic	Minimum Error (%)	Maximum Error (%)	Average Error (%)	Standard Deviation (%)
Potts-MFA heuristic	0.0	18.2	1.1	2.5
Latest Start Time(LST)	0.0	75.0	27.2	13.3
Latest Finish Time(LFT)	0.0	75.0	24.6	13.1
Minimal Slack (SLK)	0.0	35.0	12.6	9.0

XI. CONCLUSION

We proposed, developed and tested a priority rule based heuristics into Potts-MFA to resource-constrained project scheduling. Potts-MFA neural network used for serial schedule generation scheme. To the best of our knowledge, this is the first time that neural networks based heuristics have been applied to the RCPSP. So, research in this approach is still in its infancy. We tested this approach on some well-known RCPSP benchmark problem instances in the literature. The computational results are very encouraging as they compare very well with some of the best results in the literature from techniques such as priority rule-based heuristics. The approach, in spite of being relatively new, gave very good results, and therefore appears to be very promising and worthy of further exploration. Future research may focus on developing some hybrid approaches involving the Potts-MFA approach and some of the other successful approach such genetic algorithms, to further improve the results.

REFERENCES

[1] Baar, T., Brucker, P.,& Knust, S., (1998). Tabu-search algorithms and lower bounds for the resource-constrained project scheduling problem, in: S. Voss, S. Martello, I. Osman, C. Roucairol (Eds.), Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, Kluwer Academic, pp. 1– 18.

[2] Bahreininejad, A., & Topping, B.H.V. (1997). Neural Computation for structural Mechanics. Saxe-Coburg Publication, (PP. 90-137). U.K.

[3] Bell, C.E., & Han, J. (1991). A new heuristic solution method in resource-constrained project scheduling. Naval Research Logistics, 38, 315–331.

[4] Boctor, F.F. (1990). Some efficient multi-heuristic procedures for resource-constrained project scheduling. European Journal of Operational Research, 49, 3– 13.

[5] Bouleimen, K., & Lecocq, H. (1998). A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem. in: G. Barbarosoglu, S. Karabati, L. O¨ zdamar, G.Ulusoy (Eds.), Proceedings of the Sixth International Workshop on Project Management and Scheduling, Bogazici University, 19– 22.

[6] Chan, W.T., Chua, D.K.H., & Kannan, G. (1996). Construction resource scheduling with genetic algorithms. Journal of Construction Engineering and Management, ASCE 122 (2), 125– 132.

[7] Davis, E.W., & Patterson, J.H. (1975). A comparison of heuristic and optimum solutions in resource-constrained project scheduling. Management Science, 21, 944– 955.

[8] Gavish, B., & Pirkul, H., (1991). Algorithms for multi-resource generalized assignment problem. Management Science, 37 (6), 695–713.

[9] Gislen, L., Peterson, C., & Soderberg, B., (1992). Complex Scheduling with Neural Networks. Neural Computation, 4, 805-831.

[10] Hopfield, J.J., & Tank, D.W., (1985). Neural Computation of Decision in Optimization Problems. Biological cybernetics 52, 141-156.

[11] Kirkpatrik, S., Gelatt, C.D., & Vecchi, M.P. (1983). Optimization by simulated Annealing .Science, 220, 671–680.

[12] Kolisch, R., (1996). Serial and Parallel Resource-Constrained Project Scheduling Methods Revisited: Theory and Computation. European Journal of Operational Research ,90, 320-333.

[13] Lee, J.K., & Kim Y.D. (1996). Search heuristics for resource constrained project scheduling. Journal of the Operational Research Society, 47 (5), 678– 689.

[14] Leu, S.S.,& Yang, C.H. (1999). GA based multicriteria optimal model for construction scheduling. Journal of Construction Engineering and Management, ASCE 125 (6), 420–427.

[15] Padilla, E.M.,& Carr, R.I. (1991). Resource strategies for dynamic project management. Journal of Construction Engineering and Management, ASCE 111 (2), 279– 293.

[16] Patterson, J.H., (1984). A Comparison of Exact Approaches for Solving the Multiple Constrained Resource, Project Scheduling Problem. Management Science, 30, 854-867.

[17] Peterson, C., & soderberg, B., (1989). A new method for mapping optimization problems onto neural networks. International Journal of Neural Systems 1, 3.

[18] Pinson, E., Prins, C.,& Rullier, F., (1994). Using tabu search for solving the resource-constrained project scheduling problem. in: G.Barbarosoglu, S. Karabati, L. O¨ zdamar, G. Ulusoy (Eds.), Proceedings of the Fourth International Workshop on Project Management and Scheduling, Katholieke Universities Leuven, pp. 102–106.

[19] Sampson, S.E., & Weiss, E.N., (1993). Local search techniques for the generalized resource constrained project scheduling problem. Naval Research Logistics, 40, 365–375.

[20] Talbot, F.B. (1982). Resource-constrained project scheduling with time-resource tradeoffs: the nonpreemptive case. Management Science, 28 (10), 1197– 1210.

Mohammad Jaberi Hafshejani
Faculty Member of Department of Computer Engineering, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran.
E-mail: mohammadjaberi_61@yahoo.com