A SIMULATED ANNEALING ALGORITHM FOR RESOURCE ALLOCATION IN MULTIPLE PROJECTS

Neeraj Kumar

Abstract—A simulated annealing algorithm is developed in this paper for the problem of resource allocation in multiple projects. The proposed algorithm is based on two stages. In the first stage resources are allocated to the projects and given this allocation the projects are scheduled. In the second stage the resources are transferred between the projects and rescheduling of projects is done to improve the objective function. Only idle resources can be transferred from a less important project to more important project. The paper also reports the computational experience in which several multi-project problems are solved.

Index Terms—Simulated annealing algorithm, resource allocation, multi-project scheduling, and project delays.

I. INTRODUCTION

Simulated Annealing (SA), introduced by Kirkpatrick (1983), and originates from the physical annealing process in which a melted solid is cooled down to a low–energy state. In SA, instead of local search where all solutions that improve the objective function value are accepted immediately and search may terminate at local minimum, the SA algorithm attempt to avoid being trapped in a local optimum by sometimes accepting a neighbourhood move which increase the value of $f$ (objective function).

A. The Algorithm:

The pseudo code of the simulated annealing is given below:

Step(i) Read the problem data

Step(ii) Set the SA parameter: $T_{(0)}$, $S$, $N_0$, $h$, $a$

Step(iii) $T=T_{\text{max}}$

$N_0=N_0$

For 'S' steps Do

$N_i=N_i(1\text{~steps})$

For $N_i$ neighbours Do

Generate a neighbour $x'$ of the current solution $x_{\text{current}}$

Calculate $f(x')$

Calculate $\Delta = f(x') - f(x)$

If $\Delta < 0$ then store $x' = x_{\text{current}}$ and $f(x') = f_{\text{current}}$

If $f(x') < f(x)$ then store $x'=x_{\text{current}}$ and $f(x') = f_{\text{current}}$

Else if $P = \text{exp}(-\Delta/T) > X_{\text{random}}$ then store $x'$ and $f(x')$ as current

End Do $N_i$

Calculate $T = aT$

End Do $S$

Step(iii) END

II. PROBLEM DESCRIPTION

The resource allocation in multi-scheduling problem with resource transfer is defined as follows. There are $M$ projects ($i=1\ldots M$) to be executed. The number of activities in project $i$ is $|a_i|$ with $a_i$ being the terminal activity. The activities are related by the usual finish-start precedence relations. Activity $'a'$ of project $'i'$ take $d_{wi}$ units of time for its completion. There are $K$ ($k=1\ldots K$) types of renewable resources, available in quantity $R_k$. The amount of resource of type $k$ required by activity $'a'$ of project $'i'$ is $r_{ik}$. All the projects are available for simultaneous execution. These projects are independent except that they share a common pool of resource. Initially, the resources are transferred from the pool to the projects and then they may be transferred between the projects as shown in figure. The time of transfer of resources is known. An activity once started cannot be interrupted (non-preemption). The objective is schedule the projects, showing the start/finish times of each activity and the schedule of transfer of resources, that satisfy precedence and resource constraints and optimize the given performance measure.

Manuscript received January 19, 2014.

Neeraj Kumar, Assistant Prof. Mechanical Engineering, Sri Balaji college of Engineering & Technology, Jaipur, India,
A SIMULATED ANNEALING ALGORITHM FOR RESOURCE ALLOCATION IN MULTIPLE PROJECTS

III. PERFORMANCE MEASURE

The performance measure considered in this work is related to project delay. It is based on slippage of projects beyond their critical path length (figure). The measure may be expressed as total delay (summation of the delays of all the projects), and mean delay (total delay averaged over all projects). The total delay can be calculated by the expression based on the multi-project contain three project, (Kurtulus and Narula, 1985; Browning and Yassine, 2006).

IV. ASSUMPTIONS

1. The project networks are assumed in A-O-N mode with a dummy terminal activity for each project.
2. The projects are independent except that they share a common pool of resources.
3. An activity once started cannot be interrupted (non-preemption).
4. Any activity can be performed in only one manner, i.e. activities have only one mode.
5. The dummy source and the dummy sink activity have duration of zero periods and no request for any resources.
6. The resources are available at the central pool in the beginning. Once they allocate to individual project then they can not comeback to resource pool before completion of multi-project problem.
7. Allocation of resources to all projects and at any time minimum resource requirement is equal to maximum requirement of individual activity for respectively project.

Notation:

\[ x_0 = \text{initial solution} \]
\[ x' = \text{neighbourhood solution of solution (x)} \]
\[ f(x_0) = \text{initial objective function} \]
\[ \Delta = \text{difference b/w objective functions} \]
\[ T_{\text{max}} = \text{maximum temperature} \]
\[ \alpha = \text{attenuation factor} \]
\[ \text{S} = \text{step for cooling schedule} \]
\[ h = \text{parameter (used to modulate the step length.)} \]
\[ N_0 = \text{initial neighbourhood solution} \]
\[ \text{rand} = \text{random number [0…1]} \]
\[ \text{exp} = \text{exponential} \]
\[ P = \text{probability} \]
\[ S_g = \text{activities schedule set} \]
\[ D_g = \text{eligible activities set} \]
\[ \text{minLFT} = \text{minimum late finish time} \]
\[ R_{\text{idle}} = \text{idle resources} \]
\[ i,j = \text{indices for projects with } M \text{ being total number of projects. Projects numbered } 1,2,...,M. \]
\[ a = \text{index for activities in a project} \]
\[ t_{ij} = \text{transfer time units from project } i \text{ to } j \text{ (assumed to be equal for all types of resources)} \]
\[ \text{dur}_a = \text{duration of activity } a \]
V. PROPOSED ALGORITHM

In this paper a simulated annealing algorithm is proposed for the resource constrained multi-project scheduling problem with resource transfers (RCMSPPTT). As shown in figure the proposed algorithm is based on two stages. In the first stage resources are allocated to the projects and given this allocation the projects are scheduled. In the second stage the resources are transferred between the projects and rescheduling of projects is done to improve the objective function. Only idle resources can be transferred from a less important project to more important project. The SA approach is mainly used for allocation of resources to the projects. As shown in figure two schemes have been proposed for the problem.

**Figure 3: Proposed schemes**

In Scheme-1 goes through single management i.e. final solution of resource allocation to all individual projects find out by applying simulated annealing in looping of neighbourhood generation, scheduling and re-scheduling with resource.

Scheme-2 goes through dual management i.e the dual management work into two phases. Phase-1 is responsible for resource allocation to all individual projects, find out it by applying simulated annealing in looping of neighbourhood generation, scheduling . Phase-2 is responsible for Re-scheduling with resource transfer.

These algorithms have almost same features like solution representation of resource allocation and neighbourhood generation.

**Flow charts of proposed algorithm**

**Figure 4: Scheme-1**
A SIMULATED ANNEALING ALGORITHM FOR RESOURCE ALLOCATION IN MULTIPLE PROJECTS

VI. SCHEDULING OF THE PROJECTS BASED ON ALLOCATED RESOURCES

Here scheduling is taken serial schedule generation scheme shown in fig. The serial one is activity oriented and builds the schedule in n stages with n activities to be scheduled. At each stage g, an activity is selected and scheduled as early as possible such that precedence and resource feasibility is given (Lova and Tormos 2001). The minLFT rule is used for activity selection at the activity scheduling stage.

![Flow Chart of serial schedule generation scheme](image)

**Figure 6**: Flow Chart of serial schedule generation scheme

*Note*: FCFS is used as a tiebreaker but if ties are not broken by this, SOF is used as a second tiebreaker but if ties are not broken by this, the activity is selected randomly.

VII. RE-SCHEDULING WITH RESOURCE TRANSFER

We consider a project selection rule for re-scheduling of all individual projects. So, project will be selected by project selection rule and thereafter selected project will be re-scheduled.

Project selection rule is mincpl (minimum critical path length). It gives priority to shorter projects in terms of critical path length. First selected project is known as highest priority project and last selected project is known as lowest priority project.

For transfer of resources, the maximum demand of each resource for duration (D) at the receiver project is estimated. To fulfill this demand the resources are transferred from the remaining of the projects in order of their priorities (resources are transferred first from lowest priority project to higher (i.e. receiver) project so on) to the receiver project. This is done till the demand at the receiver project is fulfilled or all the sender projects are considered. The maximum amount of a resource that can be transferred from the sender project is determined by considering the maximum and the minimum demand at that project. It is important to note that the resources are not transferred from a project of higher importance to any project of lower importance. Once the
resources are transferred the activities are scheduled at the receiver project. Detail of resource transfer is given in flow chart of rescheduling with resource transfer.

Figure 7: Re-scheduling with resource transfer

Computational Study:
This computational study aims at evaluating the performance of the proposed algorithms. For this a set of test problems were generated using standard single project instances. The resource transfer times considered are of between projects.

Generation of test problems

Table: Total delay after resource transfer

<table>
<thead>
<tr>
<th>Serial No.</th>
<th>Problem</th>
<th>Scheme 1 - Delay</th>
<th>No. of Transfer</th>
<th>Scheme 2 - Delay</th>
<th>No. of Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>j001:</td>
<td>1.127345</td>
<td>879</td>
<td>7</td>
<td>882</td>
</tr>
<tr>
<td>2.</td>
<td>j001</td>
<td>1.78910</td>
<td>1276</td>
<td>6</td>
<td>708</td>
</tr>
<tr>
<td>3.</td>
<td>j001</td>
<td>5.12345</td>
<td>1255</td>
<td>11</td>
<td>1065</td>
</tr>
<tr>
<td>4.</td>
<td>j001</td>
<td>6.12345</td>
<td>900</td>
<td>12</td>
<td>1021</td>
</tr>
<tr>
<td>5.</td>
<td>j002</td>
<td>5.23456</td>
<td>2148</td>
<td>45</td>
<td>2279</td>
</tr>
<tr>
<td>6.</td>
<td>j004</td>
<td>7.89012</td>
<td>1525</td>
<td>48</td>
<td>1626</td>
</tr>
<tr>
<td>7.</td>
<td>j003</td>
<td>6.75532</td>
<td>1591</td>
<td>9</td>
<td>1625</td>
</tr>
<tr>
<td>8.</td>
<td>j003</td>
<td>3.25532</td>
<td>2103</td>
<td>227</td>
<td>1676</td>
</tr>
<tr>
<td>9.</td>
<td>j003</td>
<td>1.32534</td>
<td>1456</td>
<td>138</td>
<td>1708</td>
</tr>
<tr>
<td>10.</td>
<td>j003</td>
<td>6.89012</td>
<td>2123</td>
<td>9</td>
<td>2228</td>
</tr>
<tr>
<td>11.</td>
<td>j003</td>
<td>8.53249</td>
<td>1243</td>
<td>53</td>
<td>1246</td>
</tr>
<tr>
<td>12.</td>
<td>j003</td>
<td>9.89102</td>
<td>2123</td>
<td>152</td>
<td>2228</td>
</tr>
<tr>
<td>13.</td>
<td>j003</td>
<td>1.32546</td>
<td>1911</td>
<td>25</td>
<td>2238</td>
</tr>
<tr>
<td>14.</td>
<td>j002</td>
<td>7.18239</td>
<td>1911</td>
<td>49</td>
<td>1235</td>
</tr>
</tbody>
</table>

IX. CONCLUSIONS
This paper addresses the problem of resource allocation to multiple projects and scheduling with due consideration of inter project resource transfer times. It is assumed that initially the resources are available at a common resource pool from where they are allocated to different projects. Given the resource allocation the activities are scheduled at the projects. The idle resources at the projects may be transferred to other projects as per resource transfer schedule. Since the resource constrained project scheduling problem is considered to be a NP hard, the present problem being
generalization of the former, the present problem is also NP hard. It is thus very difficult to solve a real life problem exactly. In such a situation heuristic solution procedures, which provide good feasible solutions, are the only respite.

A Simulated Annealing based heuristic is developed for the problem of resource allocation to multiple projects and scheduling. The resource transfer times are considered explicitly. The objective considered is the minimization of total or mean delay of the projects. Two variants the algorithms are developed which are tested on a number of problems generated in this paper.

REFERENCES