Payment Scheduling Problems of Software Projects from a Bilateral Perspective

Ping Cao
School of Economic and Management, Shanghai Institute of Technology, Shanghai, China

Jian Zhang*
School of Air Transportation, Shanghai University of Engineering Science, Shanghai, China
*Corresponding author (E-mail: zhangj9860@sina.com)

Abstract
This study examines the capital-constrained payment scheduling problem related to software projects. An optimization model for the payment schedules of software projects is constructed from the joint perspective of the client and the contractor in software outsourcing projects. Then, a heuristic simulated annealing algorithm is designed to solve the model. Finally, a case study is presented to validate the model. The results show that the outsourcing benefits for both parties are closely related to the timing and amount of the payment. And both the parties could use this model to select appropriate strategies during the contract signing stage.

Key words: Payment Scheduling, Software Outsourcing, Heuristic Simulated Annealing Algorithm, Optimization Model.

1. INTRODUCTION

The project payment scheduling problem (PPSP) is an important part of cash flow management related to the objective of capital net present value (NPV) maximization (Waligora, 2014). A reasonable arrangement with regard to the timing and amount of payment is a widely discussed research topic in traditional engineering project management (Wiesemann, 2010). In this context, the objectives of the contractor and the client are in conflict, and their concerns related to cash flow as well as their preferences differ. During project execution period, both parties tend to take actions that maximize their own profit instead of reaching a compromise (Sobel, 2009). The majority in literature is from clients’ perspective (Paraskevopoulos, 2012; Kone, 2011; Kim and Jr. Ellis, 2010; Kopanos, 2014), while some are from providers perspective (Vorontsova and Rusu, 2014). Hence, payment scheduling is a key issue for both the contractor and the client. Similar to the case in traditional engineering projects, the payment scheduling arrangement of software projects has a direct impact on the profits of the contractor and the client. Whereas with the rapid growth in the number of software projects, the high failure rate of software projects has caused concern in the industry. The software industry has come to realize that many of the management techniques and methods that are commonly used in traditional engineering projects are not suitable for the management of software projects (Jorgensen, 2014). While research on software project management is imperative, the extant research on software project management is not up-to-date. Therefore, research on the payment scheduling of software projects in order to balance the profits of both parties for maximizing the overall profit of the software project has important theoretical and practical implications.

In the context of engineering projects, a recent overview of the resource-constrained project scheduling problem has been extensively discussed (Leyman and Vanhoucke, 2015). (Bahrami and Moslehi, 2013) introduced cash flow into project scheduling and constructed an optimization model of cash flow that would maximize the NPV. (He, 2014) studied multiple execution modes of the project payment scheduling arrangement and developed two heuristic algorithms—simulated annealing and tabu search—for the multi-mode PPSP. They compared the performance of these algorithms on a randomly constructed data set. Leyman presented a payment scheduling model in which the timing and amounts of payment would be determined in a single-mode resource-constrained project at activities’ completion times (Leyman and Vanhoucke, 2015). And then they extended a new scheduling technique to solve multi-mode resource-constrained project scheduling problem by employing genetic algorithm and penalty functions (Leyman and Vanhoucke, 2016). Because of the unique nature of the cash flow and the periodicity of client profits in software projects, these prior studies are not applicable to the payment conditions of software outsourcing projects, since they focus on general engineering projects. Moreover, these studies rarely considered the limits of the total amount of capital.

The cash flow of software projects differs from that of general engineering projects in specific ways (Gorla and Somers, 2014). First, the contractor in a software project includes the expenditure incurred during project development as well as the operational and maintenance expenses of the software systems. Second, the clients
can obtain profits only after the delivery and operation of the software, and they receive profits periodically. Finally, the payment scheduling arrangement of software projects usually involves certain capital constraints.

This study examines the payment scheduling problem of software projects (SPPSP) under certain capital constraints. A mathematic model for this issue is constructed with the aim of maximize profits from the individual perspectives of the contractor and the client and from a joint perspective. The operational and maintenance expenses concerning the contractor’s profits are included in the model. In the meantime, the client’s profits from different periods are also calculated with respect to a reference point according to the time cost of capital, in order to reflect the characteristics related to the client’s profits and timing in software projects. In addition, a corresponding heuristic simulated annealing algorithm is designed to solve the model. Finally, the research findings are explained with a case study.

2. RESEARCH FRAMEWORK

2.1 Basic Research Assumptions

The research method adopted in this study is based on activity. That is, the project was explained using an activity-on-arc (AOA) network. The arrow represents the activity, while the node represents the event. The basic assumptions of this study are as follows:

1) The graph of the project activity network can be drawn accurately.
2) The schedule and expenses of the software project activities are known and certain; there is a deadline agreed on by both parties.
3) The amount to be paid is decided through negotiations when the contract is signed by both parties.
4) The project payment is based on milestone events. If the project contains l milestone events, a part of the project payment has to be paid after a milestone event is completed.
5) The client’s profits are received in installments after the delivery of the software. This amount of cash flow can be discounted until the completion of the project through the cash flow within the operation period of the software after its operation begins.
6) The amount of every installment of payment is calculated according to specific ratios of the contractor’s accumulative earned value.
7) The contractor can put in different degrees of effort q. Timing and costs are different under different degrees of effort q.

2.2 Semiotic Definitions and Definitions of SPPSP

In a certain software project, the number of activities, events, and milestones are assumed to be N, M and L, respectively. The main semiotic definitions adopted in this study are as follows:

- \( ES_m \): the earliest time that event m starts
- \( LS_m \): the latest time that event m starts
- \( \eta \): distribution ratio of the activity’s expenses between the start and the end of the activity
- \( d_{ij} \): duration of project activities \((i,j)\)
- \( U \): total payment of the project contract
- \( D \): project deadline
- \( D_s \): software usage period
- \( t \): variable during the period of project execution \((t = 0,1,...,D)\)
- \( r \): variable during the period of software usage \((r = 1,...,D_s)\)
- \( B \): client’s profit in the rth installment \((r = 1,...,D_s)\)
- \( C \): client’s maintenance expenses in the rth installment

The expense of event n is \( c_n \). Therefore, \( c_m \)—the expense required to complete event m—is expressed as:

\[
c_m = \eta \sum_{n\in N_{m1}} c_{nq} + (1 - \eta) \sum_{n\in N_{m2}} c_{nq}
\]  

(1)

\( N_{m1} \) adopts event m as the start of the activity, while \( N_{m2} \) adopts event m as the end of the activity. \( c_{nq} \) is the expense related to the degree of effort put in by the contractor for q amount of time. \( \omega_n \) is the earned value of event n. Therefore, \( \omega_m \), the earned value of event m, is:

\[
\omega_m = \sum_{n\in N_{m2}} \omega_n
\]  

(2)

\( y_n \) is the 0-1 variable of whether event m takes place during period t. \( x_n \) is the 0-1 variable of whether the payment of event m is made. \( P_k(k=1,2,...,K) \) represents the amount paid at the kth installment of payment. \( B_t \) represents the client’s profits during the tth period. Both parties pursue the payment scheduling for project profit
maximization, which is represented by $\prod \pi = \{ T^*; P^* \}$, where $P^*$ is the set of the best payment amounts $P^* = \{ p_k; k=1,2,\ldots,K \}$ while $T^*$ is the set of the time intervals over which the event takes place $T^* = \{ t, y_m = 1; m = 1,2,\ldots,M \}$.

3. CONSTRUCTION OF SPPSP OPTIMIZATION MODEL

3.1. Contractor’s SPPSP Optimization Model

During the execution of the software project, it is assumed that the client has to make $K$ number of payments to the contractor ($K=L+1$), with one payment made at the start of the project, and the rest of the payments made at each milestone event. Assuming the profit after period $t$ is discounted to the profit at the start of the project as $\exp(-\alpha t) = (1 + \alpha)^{-t}$, where $\alpha$ is the discounted rate during the execution phase of the project. From the perspective of the contractor, an optimization model of SPPSP can be constructed as:

$$\max \ NPV_{\text{contractor}} = \sum_{k=1}^{K} p_k \sum_{m=1}^{M} \sum_{t \in E_m} \{ \exp(-\alpha t) y_m x_m \} - \sum_{m=1}^{M} c_m \sum_{t \in E_m} \{ \exp(-\alpha t) y_m \}$$

s.t. $\sum_{t \in E_m} y_m = 1 \quad m = 1,2,\ldots,M$,

$$\sum_{m=1}^{M} x_m = K$$

$$\sum_{t \in E_m} y_m t + d_m \leq \sum_{t \in E_m} y_m w \quad v = 1,\ldots,M-1; w = 2,\ldots,M$$

$$\sum_{k=1}^{K} (p_k x_k \sum_{t \in E_m} y_m) \leq \beta \sum_{m=1}^{M} \sum_{t \in E_m} \{ \omega_m y_m \} \quad T = 1,2,\ldots,D$$

$$\sum_{k=1}^{K} p_k = U \quad k = 1,2,\ldots,K$$

$$x_m = \begin{cases} 1 & m \in m_m \text{ or } m = 1 \\ 0 & \text{else} \end{cases}$$

$$y_m = 1 \text{ or } 0$$

The target function in equation (3) is the maximized NPV of the contractor’s cash flow; the constraint conditional in equation (4) ensures that event $m$ takes place within the time window $[E_m, L_m]$; the constraint in equation (5) represents $K$, the number of payments; equation (6) refers to the precedence constraint relationship between the start and the end of the event in event $(v,w)$. Equation (7) is the constraint of the payment amount. The amount of payment does not exceed a ratio $\beta \{ 0 \leq \beta \leq 1+\pi \}$ of the accumulative value earned by the contractor during the period $T$, $\pi$ being the contractor’s marginal value. Equation (8) ensures that the sum of the total payment corresponds to the total amount of payment designated in the project contract. Equation (9) represents that payment was made at the start of the project and at each milestone event.

3.2. Client’s SPPSP optimization model

The client’s involvement is periodic as his/her payment to the contractor is made over multiple instances during the execution of the project. His/her profit can be obtained only after the deli
dent fee of the system need to be included in the model. The client’s profit and costs at the start of the project as $\exp(-\alpha t) = (1 + \alpha)^{-t}$, where $\alpha$ is the discounted rate during the execution phase of the project. From the perspective of the client, an optimization model of SPPSP can be constructed as:

$$\max \ NPV_{\text{client}} = \{ \sum_{t=1}^{D} (B_t - C_t)(\exp(-\mu t)) - \sum_{k=1}^{K} p_k \sum_{m=1}^{M} \sum_{t \in E_m} \{ \exp(-\alpha t) y_m x_m \}$$

s.t. $\sum_{k=1}^{K} (p_k x_k \sum_{t \in E_m} y_m) \geq \beta \sum_{m=1}^{M} \sum_{t \in E_m} \{ \omega_m y_m \} \quad T = 1,2,\ldots,D$
The target function in equation (11) is the maximized NPV of the client’s cash flow. In equation (11),

\[ \sum_{r=1}^{D} (B_r - C_r)(\exp(-\mu r)) \]

represents the profit discounted at each period until the payment moment \( t = 1 \) when the project is completed. Multiplying it by \( \exp(-\alpha D) \) would discount the profit to the starting moment \( t = 0 \) of the project. The constraint in equation (12) ensures that the payment amount is not lower than a fixed ratio \( \beta \) of the contractor’s accumulative earned value. The other constraint conditions are the same as those in the SPPSP model from the contractor’s perspective.

### 3.3. Integrated SPPSP Model

The preferences of the client and the contractor with regard to payment scheduling are different. Under the fixed condition of the total value of the contract, the contractor can gain a higher NPV and obtain extra profits from early payments if each payment time is as early as possible, and the payment amount at the early stages is larger. On the other hand, if each payment time is arranged at a later stage, and the payment amount at the later stages is larger, the client would gain a larger profit. Therefore, research on this issue from the joint perspective of both parties is essential to arrive at a payment scheduling arrangement that is mutually agreeable. The jointly integrated SPPSP optimization model is expressed as:

\[
\text{max NPV} = NPV_{\text{provider}} + NPV_{\text{client}}
\]

\[ \begin{align*}
&= \sum_{r=1}^{D} (B_r - C_r)(\exp(-\mu r))\exp(-\alpha D) - \sum_{t=1}^{T} c_t \sum_{\tau=0}^{T} \exp(-\alpha \tau) y_{m\tau} \\
&\text{s.t. } \sum_{\tau=0}^{T} y_{m\tau} = 1, m = 1, 2, \ldots, M, \\
&\sum_{\tau=0}^{T} y_{m\tau} t + d_{m\tau} \leq \sum_{\tau=0}^{T} y_{v\tau} t, v = 1, \ldots, M - 1; w = 2, \ldots, M
\end{align*} \tag{13} \]

The constraint condition of this model is the same as the constraint in equation (3), and constraint condition (7) is adopted in this model.

### 4. HEURISTIC SIMULATED ANNEALING ALGORITHM FOR SPPSP MODEL

The PPSP was proven to be an NP-hard problem; the SPPSP model shows that it involves a mixed integer nonlinear programming issue, which is also an NP-hard problem. In the SPPSP, the set of payment events, i.e., which payments are made at the starting event and at the milestone event, is known. As long as the event schedule \( T^* \) and the payment amount \( P^* \) are known, the payment scheduling \( T^* = [T^*, P^*] \) can be certain. Based on the characteristics of the SPPSP, the optimal scheduling arrangement \([12, 13]\) for a software project with different payment events was calculated using the heuristic simulated annealing algorithm. The steps of the algorithm are as follows:

**Step 1:** Ascertain the initial solution \( T_0 \). Without violating the precedence relationship constraint as a precondition, arrange a realization time at random within the time window during each event to arrive at a realization time vector \( T \) of an event. If the realization time of ending event \( m \) in \( T \) is no later than the project deadline \( D_m \), \( T \) is accepted as the plausible realization time vector \( T_o \) of the initial event. Repeat this step until a plausible \( T_o \) is obtained. Calculate \( P_0 \) and profit \( NPV_0 \).

**Step 2:** The initial temperature is \( t = Temp_0 \). The ending temperature is \( Temp_0 \). The cooling speed is \( \mu (0 < \mu < 1) \). The search precision is \( \xi \). Because of the length of the Markov chain \( L, L = 10 \times N \) (\( N \) is the number of events in the project).

**Step 3:** \( T_0 \) is derived using the neighbor point \( T_1 \). Excluding the starting event, an event is selected at random from the rest of the events. Ensuring the satisfaction of the network precedence relationship as a precondition, a unit is altered at random within the time window of the realization time of the event. Whether the realization time of the ending event exceeds the deadline of the project is examined. If not, a feasible neighbor point \( T_1 \) of the present \( T_0 \) is obtained. Otherwise, the operation is repeated until a feasible \( T_1 \) is obtained. The NPV is included in the calculation of \( T_1 \).

**Step 4:** If \( \Delta NPV = NPV_1 - NPV_0 > 0 \), the neighbor point is the present solution: \( T_0 \leftarrow T_1 \). Otherwise, a random number \( R \) is derived between \([0, 1]\). If \( R \leq e^{\Delta NPV/T} \), the neighbor point is accepted as the present solution: \( T_0 \leftarrow T_1 \). Otherwise, \( T_1 \) is rejected.
Step 5: The revised temperature is $\text{Temp}_1 \leftarrow \mu \text{Temp}_0$. If $\text{Temp}_1 > \text{Temp}_f$ is satisfied, return to Step 3. If the condition is not met, the search process is stopped.

Step 6: The final results $T_0$ and $\text{NPV}_0$ are produced as the search output. The search process comes to an end.

5. CASE ANALYSIS

5.1. Project Parameters

Figure 1 depicts the AoA network for a software project that comprises 16 events and 20 activities. The number on each arc represents the serial number of the activity. This study used the case of a hypothetical project to validate the proposed model. In this hypothetical case, an enterprise is set to invest 3 million dollars for the construction of an enterprise resource planning (ERP) system; 5% of the amount is pre-payment that is to be settled upfront. After the planning and preliminary design stage, the system is expected to be operational in 80 weeks. Considering that the ERP system will be in operation for 5 years, the cash flow amounts of the profit and expenditure (operational and maintenance fee) each year after the system starts operation are shown in Table 1.

![Figure 1. Network of software project](image)

<table>
<thead>
<tr>
<th>Time $\tau$ (year)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profit $B_{\tau}$ (in $10,000)</td>
<td>70</td>
<td>95</td>
<td>125</td>
<td>130</td>
<td>130</td>
</tr>
<tr>
<td>Expenditure $C_{\tau}$ (in $10,000)</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

The distribution ratio $\eta$ of the activity expenses from the start to the end of the event is 0.5. $\beta = 85\%$. The discount rate $\alpha$ is 0.032%, which was the weekly interest rate of Bank at the time. The discount rate $\mu$ is 5.94%, which was the annual interest rate of borrowing of the bank at the time. The client’s payment to the contractor is made in five installments. The calculation results in Table 2 show the earned values of each activity, the work period, and the costs.

<table>
<thead>
<tr>
<th>Activity $n$</th>
<th>Earned value (in $10,000)</th>
<th>Work period (week)</th>
<th>Cost (in $10,000)</th>
<th>Activity $n$</th>
<th>Earned value (in $10,000)</th>
<th>Work period (week)</th>
<th>Cost (in $10,000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15.2</td>
<td>8</td>
<td>11.6</td>
<td>11.2</td>
<td>16.2</td>
<td>16</td>
<td>12.8</td>
</tr>
<tr>
<td>2</td>
<td>8.52</td>
<td>8</td>
<td>6.6</td>
<td>12</td>
<td>16.1</td>
<td>8</td>
<td>13.6</td>
</tr>
<tr>
<td>3</td>
<td>23.4</td>
<td>12</td>
<td>19.5</td>
<td>13</td>
<td>6.3</td>
<td>4</td>
<td>5.6</td>
</tr>
<tr>
<td>4</td>
<td>12.3</td>
<td>8</td>
<td>9.5</td>
<td>14</td>
<td>9.7</td>
<td>12</td>
<td>8.2</td>
</tr>
<tr>
<td>5</td>
<td>18.3</td>
<td>16</td>
<td>14.5</td>
<td>15</td>
<td>6.4</td>
<td>14</td>
<td>14.5</td>
</tr>
<tr>
<td>6</td>
<td>13.7</td>
<td>16</td>
<td>11.4</td>
<td>16</td>
<td>5.1</td>
<td>4</td>
<td>4.2</td>
</tr>
<tr>
<td>7</td>
<td>38.2</td>
<td>16</td>
<td>34</td>
<td>17</td>
<td>22.3</td>
<td>8</td>
<td>19.5</td>
</tr>
<tr>
<td>8</td>
<td>25.2</td>
<td>14</td>
<td>21.6</td>
<td>18</td>
<td>3.26</td>
<td>2</td>
<td>2.8</td>
</tr>
<tr>
<td>9</td>
<td>21.6</td>
<td>10</td>
<td>17.2</td>
<td>19</td>
<td>12.4</td>
<td>8</td>
<td>9.6</td>
</tr>
<tr>
<td>10</td>
<td>17.4</td>
<td>8</td>
<td>15.2</td>
<td>20</td>
<td>3.64</td>
<td>3</td>
<td>3.2</td>
</tr>
</tbody>
</table>

5.2. Results

The time window $[E_{\text{min}}, L_{\text{max}}]$ in which each event takes place was calculated using the known data in Table 2, and the costs of each event ($c_{\text{e}}$) and the earned value ($o_{\text{e}}$) were calculated using equations (1)–(2). Five milestones were set for the project, and payments were made at each of the milestone events. The prepaid amount of 5% was paid at the start of the project; therefore, only 95% of the contract price (2.85 million dollars) was paid at the five milestone events. The optimal payment scheduling $\prod^* = \{T^*, P^*\}$ from the different
perspectives and at the different milestone events (the sets of milestone events are \{2,6,11,15,16\} and \{3,9,12,14,16\}) are listed in Table 3.

<table>
<thead>
<tr>
<th>Payment event set</th>
<th>Perspective</th>
<th>Optimal payment scheduling</th>
<th>NPV_{prev}</th>
<th>NPV_{client}</th>
<th>NPV_{PC}</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,6,11,15,16</td>
<td>contractor</td>
<td>$P^*=(12,92,109,16,59,93,48,45,54,55)$</td>
<td>14.71</td>
<td>73.05</td>
<td>87.76</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T^*=(0,8,20,36,36,36,26,46,36,54,61,75,69,77,80)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>client</td>
<td>$P^*=(12,92,62,82,92,58,62,14,5)$</td>
<td>12.04</td>
<td>76.37</td>
<td>88.41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T^*=(0,8,20,40,40,36,42,26,66,54,61,75,69,77,80)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Joint</td>
<td>$P^*=(12,92,62,82,92,58,62,14,5)$</td>
<td>12.47</td>
<td>76.29</td>
<td>88.76</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T^*=(0,8,20,40,40,36,42,26,66,54,61,75,69,77,80)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3,9,12,14,16</td>
<td>contractor</td>
<td>$P^*=(43,27,102,53,65,71,23,20,30,21)$</td>
<td>13.67</td>
<td>57.04</td>
<td>70.71</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T^*=(0,8,20,36,36,36,26,46,36,54,61,75,69,77,80)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>client</td>
<td>$P^*=(32,81,99,3,65,71,32,64,54,55)$</td>
<td>13.11</td>
<td>59.79</td>
<td>72.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T^*=(0,8,20,45,40,40,36,42,26,66,54,61,69,77,80)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Joint</td>
<td>$P^*=(12,92,62,82,92,58,62,14,5)$</td>
<td>10.64</td>
<td>62.39</td>
<td>73.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T^*=(0,8,20,40,40,36,42,26,66,54,61,75,69,77,80)$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.3. **Analysis**

The results presented in Table 3 indicate significant differences between the contractor’s and the client's net cash flow and desirable payment scheduling. In this example, the contractor's maximized profit is $NPV_{prev} = 14.71$, and the client's profit is the lowest ($NPV_{client} = 73.05$). In contrast, later payment points and larger amounts of payment at later payment events would lead to greater profits for the client. When the client's maximized profit value is 76.37, the profit of the contractor is lower (12.04). Thus, the following conclusion could be drawn: under the condition of fixed payment amounts and payment points (milestone events), the profits of both outsourcing parties are related to the payment point and the payment amount. Regarding the arrangement of the payment amount, the contractor usually prefers the earliest possible payment time and hopes to settle larger amounts of payment at an earlier payment points. Hence, if the aim is to maximize the contractor's profit, the contractor should consider reducing the project costs as the higher priority. If the aim is to maximize the client's profit, the client should consider obtaining greater than expected profit at the end of the project as early as possible to be a high priority and reducing project costs as a low priority. Therefore, when deciding the payment scheduling for software projects, both outsourcing parties should balance each other’s profits and determine a mutually agreeable payment schedule.

6. **CONCLUSION**

This study examined the optimization of payment scheduling in the context of software projects. Under certain capital constraints, optimization models for payment scheduling related software outsourcing projects were constructed from the perspectives of the client and the contractor in the project, and from the joint perspective. The proposed models included the unique characteristics of the payment events and the cash flow of a software project. By using the event-based method, Profit maximization models were constructed from the perspectives of the contractor and the client. Additionally, a profit maximization model from the joint perspective of both parties was constructed to balance the profits of the two parties. The proposed models reflect the characteristics of software project payment. Based on the characteristics of the proposed models, a heuristic simulated annealing algorithm was designed to solve the models. Finally, the payment from three perspectives was calculated using a hypothetical case study to obtain the optimal payment scheduling. The results of the calculation showed that the optimal payment stages vary depending on the different perspectives. For instance, the optimal payment schedule for a contractor is the exact opposite of the optimal payment schedule for a client. This study indicates that describing the contractor’s profit using the net cash flow can accurately reflect the real profits of the contractor, the client, and both parties together. Both the parties involved in software outsourcing could use this model to select appropriate strategies for negotiations during the contract signing stage.

It should be noted that the study in this paper is based on the assumption that the event occurrence times can be arranged without considering resource constraints. However, this is not the case in many practical circumstances where the resource availability for the contractor is limited. Therefore, SPPSP under resource constraints may be a worthwhile problem for further investigation in this direction.

**Acknowledgement**

This work was supported by the national fund of social science (10CGL005).
REFERENCES


