Contemporary Engineering Sciences, Vol. 7, 2014, no. 20, 931 - 938
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ces.2014.48114

The Interval Method of Cost Planning and Its Implementation in the Online Service

Alexander V. Ilyin and Vladimir D. Ilyin

44 Bld 2, Vavilova St.
Institute of Informatics Problems of the Russian Academy of Sciences
Moscow, 119333, Russia

Abstract

The article presents the innovative method of cost planning with taking into account the priorities of expense items, and the active online service which enables the method usage in a variety of client applications. Widest area of the service usage is the budgeting, although the service allows flexible and dynamic planning of any resource costs. The important innovation for users is ability to specify the expected resource amount and the requests for resource as numerical segments, receive results in the same form, and then clarify the plan in the course of its implementation.

Keywords: cost planning, online service, resource allocation, SaaS

1 Introduction

Income is a variable for millions of individuals and companies. It may depend on the volume of sales, market prices, exchange rates and many other factors. However, even on a state level, expenditures are often planned on the basis of point assumptions about the total income. If such assumptions are wrong, the budget is to be altered. The planning results are also presented by exact value for each expense item, although in practice most of the costs can not be predicted accurately. To address these shortcomings, principles of the interval cost planning were invented [4].

The problems of *non-priority and priority cost planning* have informal statements which contain *mandatory and orienting rules* [3]. The mandatory rules include restrictions on the resource spending (they guarantee the feasibility of solution) and rules of irredundant satisfaction of the requests for resource. Orienting rules determine the direction of search for a solution. The solution found by using the proposed algorithms, always satisfies the mandatory rules and satisfies orienting rules as much as the interval specificity of the problem allows. If fulfillment of orienting rules is possible, the solution corresponding to them is treated as more effective than any other.
The problem statement and algorithm of priority planning suggested in [4] and [2] do not reduce to non-priority planning in the case of equal priorities. The unified method of cost planning, which is described in the next section, eliminates this shortcoming.
Traditional approach, describing scheme and principles of cost planning [1], does not describe problem formulation and method of solution. The interval method and the online service have no known analogues.

2 The interval method of cost planning

2.1 The non-priority problem

For a numeric segment $[a, A] (a \ge 0, A > 0)$, which expresses the expected resource amount, and segments $[b_i, B_i] (b_i \ge 0, B_i > 0, i = 1 \ldots n)$, which express requests of expense items, a cost plan $[x_i, X_i] (b_i \ge x_i \ge 0, B_i \ge X_i \ge 0, i = 1 \ldots n; \sum_{i=1}^{n} x_i \le a, \sum_{i=1}^{n} X_i \le A)$ needs to be found. Depending on presence of the resource shortage for sum of the left bounds and sum of the right bounds of requests, we have one of the four cases:

$$(1) \quad \sum_{i=1}^{n} b_i > a, \quad \sum_{i=1}^{n} B_i > A$$

In this case, we solve *problem for the left bounds*, and then *problem for the right bounds*.

$$(2) \quad \sum_{i=1}^{n} b_i \le a, \quad \sum_{i=1}^{n} B_i > A$$

In this case, we set $x_i = b_i$ ($i = 1 \ldots n$), and then solve *problem for the right bounds*.

$$(3) \quad \sum_{i=1}^{n} b_i > a, \quad \sum_{i=1}^{n} B_i \le A$$

In this case, we set $X_i = B_i$ ($i = 1 \ldots n$), and then solve *problem for the left bounds*.

$$(4) \quad \sum_{i=1}^{n} b_i \le a, \quad \sum_{i=1}^{n} B_i \le A$$

In this case, we set $x_i = b_i$, $X_i = B_i$ ($i = 1 \ldots n$), and the problem is solved.

Problem for the left bounds is elementary. For zero left bounds ($b_i = 0$) we set

$x_i = 0$. Let I be the set of indices i, for which $b_i > 0$. The mandatory rules of planning: $\frac{x_i}{b_i} = \frac{x_j}{b_j}\ (\forall\ i,\ j \in I),\ \sum_{i\in I} x_i = a$.

The decision is $x_j = \frac{b_j}{\sum_{i\in I} b_i} a \quad (\forall j \in I)$.

Problem for the right bounds often seems to be similar and decidable by formulas $X_j = \frac{B_j}{\sum_{i=1}^{n} B_i} A \quad (j = 1\ldots n)$. Here is a simple counterexample. Let [3, 4] be the segment [a, A] and [1, 5], [4, 5] – two requests for resource. We get

$x_1 = \frac{1}{1+4} \cdot 3 = 0{,}6;\ \ x_2 = \frac{4}{1+4} \cdot 3 = 2{,}4$.

Application of $X_j = \frac{B_j}{\sum_{i=1}^{n} B_i} A \quad (j = 1\ldots n)$ makes $X_1 = X_2 = \frac{5}{5+5} \cdot 4 = 2$, that is, we got $X_2 < x_2$ (absurd).

In fact, the problem is solved by means of iterations. The mandatory rule of resource allocation is $\sum_{i=1}^{n} X_i = A$. If I is the set of indices i where $B_i > b_i$, and K is the set of indices i where $B_i = b_i$ ($I = \{1, \ldots, n\} \setminus K$), then orienting rules of allocation are

$$\frac{X_i - x_i}{X_j - x_j} = \frac{B_i - b_i}{B_j - b_j} \quad \textit{for}\ \forall\ i,\ j \in I \quad \textit{and} \quad \frac{X_i}{X_j} = \frac{B_i}{B_j} \quad \textit{for}\ \forall\ i,\ j \in K$$

2.2 *The unified problem with taking into account the priorities of expense items*

For a numeric segment $[a, A]\,(a \ge 0, A > 0)$, which expresses the expected resource amount, segments $[b_i, B_i]\,(b_i \ge 0, B_i > 0, i = 1\ldots n)$, which express requests of expense items, and weighting coefficients (priorities) of expense items $c_i > 0\ (i = 1...n)$,

a cost plan $[x_i, X_i]\,(b_i \ge x_i \ge 0,\ B_i \ge X_i \ge 0,\ i = 1\ldots n;\ \sum_{i=1}^{n} x_i \le a,\ \sum_{i=1}^{n} X_i \le A)$ needs to be found.

Problem for the left bounds

For $b_i = 0$ we set $x_i = 0$. Then the problem is solved by means of iterations.

The mandatory rule of allocation is $\sum_{i=1}^{n} x_i = a$.

Let introduce the set I of indices i, for which x_i are still not found: $I = \{i \mid b_i > 0, 1 \le i \le n\}$.

The orienting rule is $\frac{x_i}{x_j} = \frac{b_i}{b_j} \cdot \frac{c_i}{c_j}$ *for* $\forall\, i,\, j \in I$.

Let introduce the variable $\Delta a := a$ (hereinafter $:=$ is an assignment operator).

In each iteration for the left bounds which are still not found we set

$$x_j := \frac{b_j c_j}{\sum_{i \in I} b_i c_i} \Delta a \quad (j \in I)$$

Then we introduce the auxiliary set $I' := \{i \in I \mid x_i \ge b_i\}$. For each $i \in I'$ x_i is found: $x_i = b_i$. In the case of empty I' we set $I' := I$.

Then we change I and Δa: $I := I \setminus I', \quad \Delta a := \Delta a - \sum_{i \in I'} x_i$.

Now, if I is empty, the problem is solved.

Problem for the right bounds

The mandatory rule of allocation is $\sum_{i=1}^{n} X_i = A$.

Let introduce the set I of indices i, for which $b_i < B_i$, and the set K of indices i, for which $b_i = B_i$ $(I = \{1, \ldots, n\} \setminus K)$.

The orienting rules are

$$\frac{X_i - x_i}{X_j - x_j} = \frac{(B_i - b_i)c_i}{(B_j - b_j)c_j} \quad \textit{for } \forall\, i,\, j \in I \quad \textit{and} \quad \frac{X_i}{X_j} = \frac{B_i c_i}{B_j c_j} \quad \textit{for } \forall\, i,\, j \in K$$

If all priorities are equal to some constant c, it is easy to see that the problem is identical to the non-priority one.

Let introduce the variable $\Delta A := A - (x_1 + \ldots + x_n)$.

In each iteration for $j \in I$ we set

$$X_j := x_j + \frac{(B_j - b_j)c_j}{\sum_{i \in I}(B_i - b_i)c_i} \Delta A.$$

Then we introduce the auxiliary set $I' := \{i \in I \mid X_i \ge B_i\}$. For each $i \in I'$ X_i is found: $X_i = B_i$. In the case of empty I' we set $I' := I$.

Then we change I and ΔA as follows: $\Delta A := \Delta A - \sum_{i \in I'}(X_i - x_i), \quad I := I \setminus I'$.

Now, if I is empty, we stop iterations.

After the end of iterations it is possible that $\Delta A > 0$. This can happen only if for all i, where $B_i > b_i$, we have got $X_i = B_i$ (otherwise iterations would be continued, and ΔA would increase the bounds where $X_i < B_i$). In this case we distribute ΔA between expense items, for which $b_i = B_i$.

In each iteration for $j \in K$ we set

$$X_j = x_j + \frac{B_j c_j}{\sum_{i \in K} B_i c_i} \Delta A$$

Then we introduce the auxiliary set $K' := \{i \in K \mid X_i \geq B_i\}$. For each $i \in K'$ X_i is found: $X_i = B_i$. In the case of empty K' we set $K' := K$.
Then we change K and ΔA: $\Delta A := \Delta A - \sum_{i \in K'} (X_i - x_i), \quad K := K \setminus K'$.
Now, if K is empty, the problem is solved.

3 Online service and its client applications

The described method is implemented in the online service "Cost Planning".

3.1 Principles of work

Suggested concept of online services is similar to the SaaS (Software as a Service) concept [5-7]. The difference is that we do not store data of users' tasks on server and offer special client applications for Microsoft Windows®, Apple Mac OS X® and other operating systems - instead of Web applications. The second difference is explained by the fact that the stability and performance of Web applications depends on the browser, in which they work.
The basic principles are:

- Calculation algorithms are implemented in the server applications (services), which work on reliable and high-performance servers 24 hours a day, 7 days a week.
- User downloads an installer of client application (corresponding to his/her operating system) for a service and launches it. The installation process is very simple and does not require any special skills.
- A client application provides a familiar GUI for data input, file operations, etc. Only when calculations are necessary, a client application at user's command connects to a Service and sends it a request having a special text format. Request contains only numbers which are necessary for calculations - without semantic bindings, that is confidential data, such as names of resources, expenses, measure units, etc, are not transmitted thru the Internet. Request header consists of encoded data about the user's workplace (when user launches an application the first time, the workplace registration is executed).
- Service receives the request, performs parsing, makes calculations, and sends the results to the client application. None of transmitted data are stored on servers: it would be just unnecessary disk usage. All the data are stored on client side. Service processes request in operating memory, sends the reply, and terminates connection. The only thing stored on the service side - is the database containing registration data of users' workplaces. It is necessary to authenticate requests using their headers.

3.2 Advantages of online services

Use of online services gives the following advantages:

- Resource planning systems, implemented as stand-alone applications, are usually quite expensive. Use of online services is significantly cheaper. User pays only for the periods, when a service is actually used.
- The cost of service includes technical support and all updates of the client applications. User just needs to go thru elementary process of client app installation and registration of his/her workplace in the service. The hardware and software of the Service is the developer’s concern. (Stand-alone applications for resource planning are usually resource demanding, and often require paid tuning).
- Services can be used by software vendors in their own applications. A special API can be provided to interact with a service.
- Computational algorithms are implemented only on servers, eliminating unauthorized use.

3.3 Work with client application

Client application of the “Cost Planning” service is the tool for distribution of expected funds between expense items, but not one more accounting program, where "planning" is often understood as entering an income and further addition of desirable expenses until the balance is broken. Next - the main features of the client application.

User specifies an interval for expected funds assuming the worst and the best conditions. User can specify any name and measurement unit for the resource (USD, thousand euro; tonnes, etc.) and *the applied precision* for planning. The applied precision is the minimum significant resource amount - from 0.0001 to 1 billion. All the data will be rounded to that number. This allows, in particular, to solve the integer problems.

User specifies a table of expense items, and for each row the lowest and the highest expected costs (or exact value) can be entered. These are requests of expense items.

User can create a separate table of details for each expense item. Amount of the resource allocated to the expense item will be distributed between expense items that form its details. The number of detail levels for expense items is not limited, so *a hierarchy* of resource consumption can be defined.

User can specify weighting coefficients (priorities) of expense items and use them in calculations.

For each table of expense items in a hierarchy, user can set its own flag 'Use weighting coefficients' and applied precision (restriction: a table data can not be more accurate then their details).

Some requests can be marked as obligatory (e.g. mortgage payments or rents can rarely be reduced).

User commands 'Allocate' from a client application. It connects to the Service via Internet and sends it a request for resource planning. The Service performs calculations and sends results back to the client application, which shows them to user. The results contain values 'Give min.' and 'Give max.' for each expense item. (Later the 'Allocate' command may be executed for the entire hierarchy or a part of it).

Sums of minimum and maximum ‘Give’ values comply the specified minimum and maximum funds respectively, and the values comply the specified requests for resource and (optionally) the priorities of expense items.

Subsequently, when user receives or spends a part of the funds, or obtains more precise information on the income or expenditure sides of the budget (let say in terms of budget planning), he/she adjusts the input data, commands 'Allocate' again, and gets refined results.

When user specifies the funds exactly (i.e. minimum = maximum) and command 'Allocate', the received values 'Give max.' can be treated as exact decision of the cost planning task.

Thus, if user specifies minimum and maximum funds and costs cautiously and follows the plan, he/she will always stay within the budget. User sees in advance how resource can be allocated; user can look at different variants of allocation and adjust expense items; user clarifies plan in the course of its implementation.

If the planning results are too "tight", user can temporarily exclude any expense item from consideration: it is enough to just put a "tick" in the corresponding cell of a table.

User can manually adjust the planning results. The application will notify if the entered data is inconsistent.

3.4 Notes on implementation

The online service, in addition to the method of cost planning, implements a unique algorithm to modify the calculation results in accordance with applied precision set at different levels of the allocation hierarchy. Description of the algorithm is out of scope of this article.

4 Conclusion

Circle of people, for whom the method implemented in the online service “Cost Planning” is useful, - from individuals and businessmen who plan their budgets - to experts on the distribution of corporate and government resources. Software vendors can use the service in their own applications.

Online services implementing the technology, which was described in [3], are being developed.

The services and client applications are presented at www.res-plan.com.

References

[1] AACE® International, Total Cost Management Framework. 2012. http://www.aacei.org/non/tcm/TCMFramework_WebEdition.pdf

[2] A.V. Ilyin, Ekspertnoe Planirovanie Resursov [Expert Resource Planning]. IPI RAN [Institute of Informatics Problems of the Russian Academy of Sciences], Moscow, 2013.

[3] A.V. Ilyin, V.D. Ilyin, The technology of interactive resource allocation in accordance with the customizable system of rules. *Applied Mathematical Sciences*, Vol. 7, 2013, no. 143, 7105-7111. http://dx.doi.org/10.12988/ams.2013.311649

[4] V.D. Ilyin, Y.V. Gavrilenko, A.V. Ilyin, E.M. Makarov, Matematicheskie sredstva situatsionnoy informatizatsii [Mathematical Tools for Situational Informatization]. Nauka, Fizmatlit, Moscow, 1996.

[5] K.A. Jamsa, Cloud Computing. Jones & Bartlett Learning, Burlington, 2013.

[6] M.J. Kavis, Architecting the Cloud. John Wiley & Sons, Hoboken, New Jersey, 2014.

[7] Trumba Corporation, Five benefits of Software as a Service. 2007. http://www.trumba.com/connect/knowledgecenter/pdf/Saas_paper_WP-001.pdf

Received: August 1, 2014