

Policy Constraint Wizard How-To: Optimal Capital Budgeting

Zero-One Portfolio Optimization Formulation

Zero-one programming is constrained optimization method that is capable of solving a large-scale complex capital investment problem. The term 'Large scale' is used to convey the concept of a large number of projects (perhaps numbering into the hundreds). Similarly, the term 'Complex' is used to indicate a decision-model where not all projects are independent (hence, ranking methods cannot be applied directly). When projects exhibit dependencies including time delays, the problem is characterized by a complex structure.

The objective of the optimization model is to maximize the capital investment portfolio's contribution to firm value while observing all constraint limitations associated with the identified dependencies.

Identifying the optimal mix of projects in a large-scale capital budgeting problem is often presented as a mathematical programming problem where the final decision variables are restricted to solution values of zero (0) or one (1). One indicates a selection recommendation where zero is taken as a rejection of the project.

WinORS automatically sets-up the basic zero-one model formulation as part of the capital budgeting solution process. The problem is always established on the spreadsheet tab following the basic capital budgeting problem; in our sample case, the model is on **Tab B**:

A Zero-One Model Example

	A	I	J	K	L	M
1		Project 8	Project 9	Project 10	Sense	RHS
2		I	I	I		
4	MAX NPV	-4557.142	3927.184	8852.594		
6	Expense	30099.563	3923.292	18551.100	≤	341890.523
7						
8						

Presented above is a small example of what the WinORS zero-one capital investment problem will look like on Tab B (Zero One). Note that the LOCK button was toggled to the ON condition to show this screen. Note how the NPVs are automatically inserted across row 4 of the spreadsheet. The variable types are set to "I" for the required solution method (on row 2). Row 5 is a narrow separator row (required in canonical form mathematical programming applications). Finally, row 6 presents a constraint. This constraint is hypothetical and inserted for visual reference only. As entered the constraint is a budget constraint that has copied the year 0 expense for each project and related to the less-than-or-equal to amount (sense column) of the right hand side (RHS). The RHS is computed (in this case) by a formula (see cell M6).

Adding Managerial / Contingent Policy Constraints

The zero-one model is an effective method for treating contingency constraints. When projects are not truly independent (the adoption of one project has no bearing on any other project), then it may become necessary to model important model the interrelationships among dependent projects. To reach the policy wizard follow the menu tree: /AFCC.

The screenshot shows the 'Constraint Wizard' dialog box. On the left, under 'Types', the 'Mutually Exclusive' radio button is selected. The 'Required Project(s)' list contains seven entries: [B] Project 1, [C] Project 2, [D] Project 3, [E] Project 4, [F] Project 5, [G] Project 6, and [H] Project 7. The 'Conditional Project' list is currently empty. At the bottom, there are three buttons: 'OK' (with a green checkmark), 'Cancel' (with a red X), and 'Help' (with a question mark).

Mutually Exclusive

Mutually exclusive projects occur when the adoption of one project precludes the adoption of another. For example, if project 1 and 3 are mutually exclusive, the following constraint would be required as part of the zero-one formulation:

$$X_1 + X_3 \leq 1$$

The WinORS policy constraint wizard should be completed as follows:

This screenshot shows the 'Constraint Wizard' dialog box after configuration. The 'Types' section remains with 'Mutually Exclusive' selected. In the 'Required Project(s)' list, the checkboxes for '[B] Project 1' and '[D] Project 3' are now checked. The 'Conditional Project' list remains empty. The 'OK', 'Cancel', and 'Help' buttons are still present at the bottom.

Upon selection of OK, the wizard will add the appropriate constraint and adjust the data range to include this constraint. Note: the sense of the added constraint is of the less-than-or-equal-to type. Hence, it is possible to select either project or neither project. To force the adoption of at least one project manually change the sense of the constraint to an equality sense. Be sure to: a) position your cursor in the sense column on the correct row and then b) use the symbol drop down list at the top of the spreadsheet.

Contingency: B Given A

Also stated as: if project A is accepted then project B must be accepted.

Often, management is faced with the requirement to accept one project if some other project is accepted. For example, management may be considering a choice of new computers. The choice is between computers that use the Microsoft Windows operating system versus computers that rely upon the Linux operating system. The choice of software applications to purchase will depend on which computer operating system is selected. To completely specify the zero one optimization model, the following constraint is required. Assume that project 3 is the proxy for the selection of the Linux computer (hardware decision). Further assume that project 1 is the proxy for Linux based application software. The appropriate policy constraint to force the adoption of Linux compatible software given the selection of Linux-based hardware (computers) is: $X_1 - X_3 = 0$
The WinORS policy constraint wizard should be completed as follows:

The screenshot shows the 'Constraint Wizard' dialog box with the following configuration:

- Types:**
 - Mutually Exclusive
 - P(i) if P(k) is Accepted
 - At Least n from k
 - No more than n from k
 - Accept P(i) only if P(i) and P(k)
 - Time Delay P(i)
 - Conditional Combinations
- Required Project(s):**
 - [B] Project 1
 - [C] Project 2
 - [D] Project 3
 - [E] Project 4
 - [F] Project 5
 - [G] Project 6
 - [H] Project 7
- Conditional Project:**
 - [B] Project 1
 - [C] Project 2
 - [D] Project 3
 - [E] Project 4
 - [F] Project 5
 - [G] Project 6
 - [H] Project 7

Buttons:

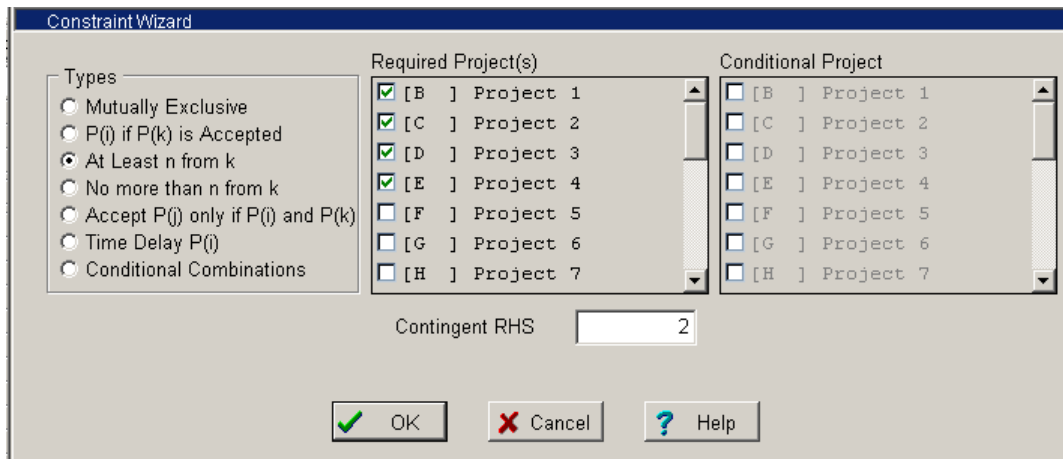
Upon selection of OK, the wizard will add the appropriate constraint and adjust the data range to include this constraint.

Contingency: Accept K From N

This type of contingency requires a certain number of projects (K) to be accepted from among an identified group (N). For example, it may be necessary to accept at least 2 plant locations from among 4 alternatives. The constraint added to the zero-one formulation is of the form:

$$X_1 + X_2 + X_3 + X_4 \geq 2$$

The WinORS policy constraint wizard should be completed as follows:



Note that the input box titled 'Contingent RHS' where RHS means the right-hand-side of the constraint equation has an entry of 2. This constraint requires at least 2 projects from among 1 through 4 to be selected.

Contingency: No More than N From Among K Projects

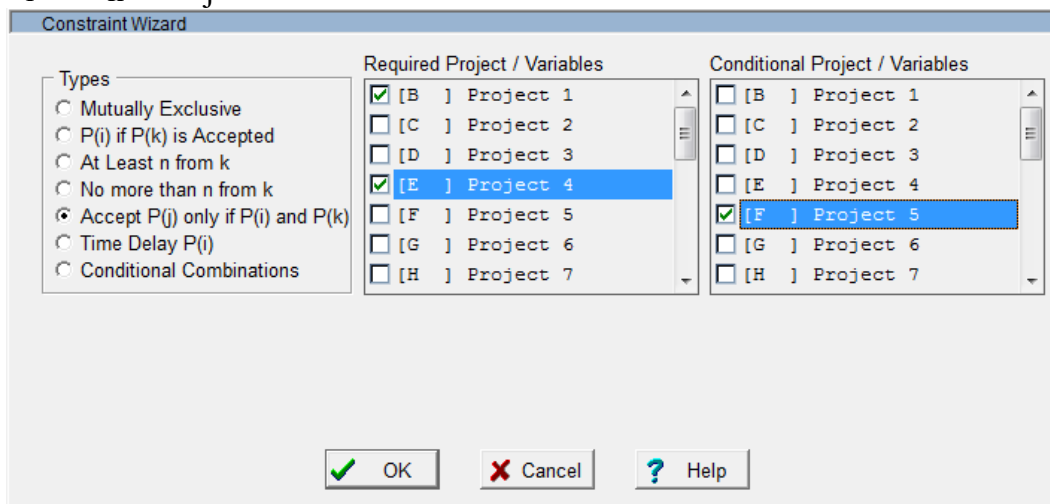
The effect of constraint choice titled 'No more than n for k' is used to limit the number of projects selected from those currently under k' consideration. Using the exhibit in the prior example, the choice of this radio button would have the effect of limiting the choices to no more than 2 from among the four available choices. Upon selection of OK, the wizard will add the appropriate constraint and adjust the data range to include this constraint.

Contingency: Accept J only if both i and k are accepted

This contingency is used for the case where two projects, A and B, must be accepted before it is possible to accept a third project – project K. Stated differently, project K can only be accepted once both projects A and B are accepted. The constraint structure is stated as follows:

$$(i + k)X_j \leq X_i + X_k , \text{ or } 2X_j \leq X_i + X_k , \text{ or}$$

$$X_i + X_k - 2X_j \geq 0$$

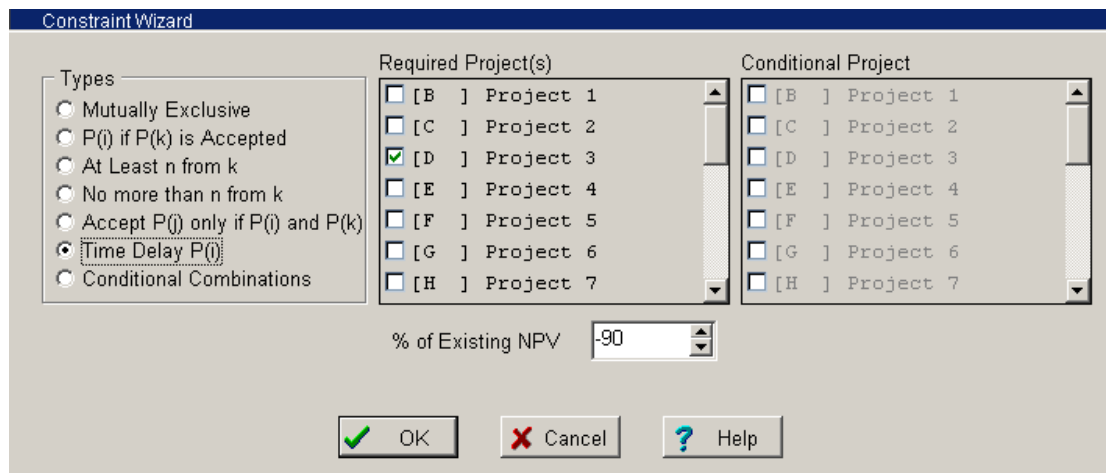


Contingency: Time Delay

Do we accept the project in the current year or delay its acceptance by one year? This scenario is often complicated by the fact that the delay results in a decay of the computed NPV. For example, consider the scenario where project 3 can be accepted in the current year or delayed by one year with a 10% decay in the computed NPV. This contingency requires both a constraint and a new variable. For example, if there are 20 projects under consideration, a new project (21) is added to the problem formulation. The objective function value for this new project is a value equal to 90% (a 10% reduction) of the project 3 NPV ($NPV_3 * 0.9$). Finally, a new constraint is added. The constraint is a mutually exclusive formulation between project 3 and the new project 21.

$$X_3 + X_{21} \leq 1$$

The WinORS policy constraint wizard should be completed as follows:



After selection of OK the wizard will add a new column to the spreadsheet formulation. Simultaneously, the wizard will insert an objective function value is 90% of that for project 3. All range settings are updated automatically.

Contingency: Combine Projects *i* and *k* along with projects *j* and *p* Into Two New and More Efficient Projects, *g* and *h*.

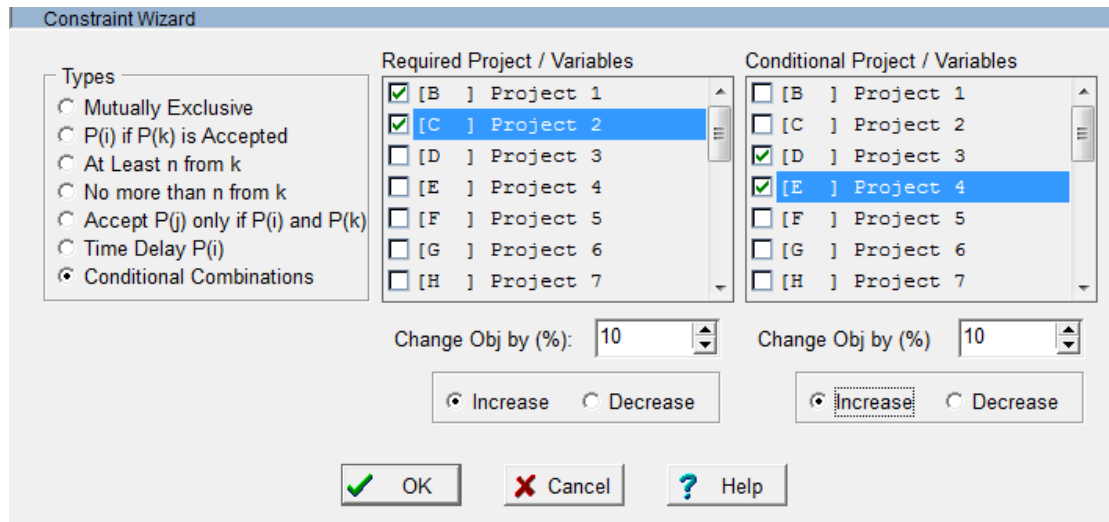
Projects *i* and *k* and projects *j* and *p* can be combined into complementary or composite projects wherein NPV values of the new combined projects will increase by 10% compared with the total of the separate projects. In this solution, the size of the problem is increased by two columns (projects) where each now represents project *g* and *h*, respectively.

- a) Define two new variables: *g* and *h*

$$X_i + X_k + X_j + X_p + X_g + X_h \leq 1$$

$$X_g + X_h \geq 1$$

- b) Adjust objective function NPV coefficients for projects *g* and *h* as required.



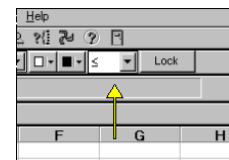
How to Solve for the Optimal Investment Plan

This example assumes that you have already employed the policy wizard to enter all policy constraints. Generating a solution to the complex capital investment decision is simply a matter of using the following menu tree: /AFCO. This sequence will require that you save your data to disk prior to generating a solution. Only use this method if you have entered all constraints via the policy constraint wizard. If you entered any constraint manually, then follow the instructions in the next section.

Manual Settings for the Optimal Investment Plan

To solve the zero-one capital investment problem follow these simple steps:

1. Enter any additional constraints on subsequent rows as dictated by the model formulation. Additional constraints begin on row 7 as labeled. Remember, a blank cell is treated as a zero coefficient in a mathematical programming model. For example, if the constraint requires a coefficient of one (1.0) for variables 1 and 3, then the second project (variable) can have either a blank cell or an entry of zero.
2. Move the cursor to the column that contains the sense of the constraint. Use the SYM button to insert the appropriate sense of the constraint.
3. After all constraints have been entered, optimize the capital budget by executing the following menu commands: /AFCO. Use standard range concepts to set the block of cells that define the decision problem.
4. For example, the following model has one constrain, ten projects, the sense column, and the RHS column. Set a solution range of B4:M6.
5. If you add a constraint, say on row 7, the next time you choose to optimize the solution, be sure to increase your range to cover the newly added constraint (e.g., B4:M7).



The Zero-One Solution Screen

A typical zero-one solution screen is presented below (this example screen is not necessarily related to any other example). The solution screen reports: a) the *number of iterations* used by the algorithm to reach an optimal solution (generally, not of importance to capital investment problems); b) the *objective function value*; and, c) the solution for each variable (either zero or one).

Remember, a solution value of one (1.0) is an affirmative selection; by contrast, a solution value of zero (0.0) is a rejection.

The *objective function value (Obj. Value)* is the sum of the net present values associated with all selected projects (those with a solution value of one). This value is automatically transferred back to the original data spreadsheet and seen as part of the **Cumulative Benefits** display.

CIE16		
Solve: ZeroOne		BIB4:M6
	A	B
1	OPTIMAL	
2	Iterations	11.000
4	Obj. Value	244401.000
5		
6	VARIABLE	VALUE
7	Project_1	1.000
8	Project_2	1.000
9	Project_3	0.000
10	Project_4	0.000
11	Project_5	0.000
12	Project_6	1.000
13	Project_7	1.000
14	Project_8	1.000
15	Project_9	0.000
16	Project_10	0.000
17		

The optimal value (244401, or \$244,401) is automatically transferred back to the spreadsheet tab with the option template. The value is pasted in the 'cumulative benefits' section of the spreadsheet.

The post is made in order to facilitate a comparison of the optimal portfolio with the portfolio obtained by ranking (using any of the ranking measures -- NPV, IRR, etc.). Compare the composition of the portfolios at approximately the same portfolio value of \$244,401. The benefits of the constrained optimization solution should become immediately obvious.

Last Update: 22-Jan-07; 24-May-12