TIME, COST AND PERFORMANCE TRADEOFFS IN PROJECT MANAGEMENT

By

LUIS F. COPERTARI

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

McMaster University

TIME, COST AND PERFORMANCE TRADEOFFS IN PROJECT MANAGEMENT

DOCTOR OF PHILOSOPHY (2002) McMaster University
(Management Science/Information Systems) Hamilton, Ontario

TITLE:	Time, Cost and Performance Tradeoffs in Project Management.
AUTHOR:	Luis F. Copertari
SUPERVISOR:	Dr. Norman P. Archer
NUMBER OF PAGES:	xi, 115

Abstract

Successful project selection and management requires optimal supervision of corporate resources within specifications for time, cost, and performance. We developed a model and algorithm to support decisions on these three dimensions for project managers. It combines the advantages of the Program Evaluation and Review Technique (PERT) and the Critical Path Method (CPM). Our methodology leads to more accurate results than PERT/CPM, which typically results in optimistic planning due to less than actual completion time estimates that do not consider the possibility of more than one longest (critical) path. We also estimate performance measured by the Internal Rate of Return (IRR) of the project and the tradeoffs between time/cost and performance. We allow decision makers to calculate the probability that each activity will be critical, an indication of their relative importance for managerial purposes, in polynomial time. Furthermore, our methodology provides the means to obtain the optimal time/cost schedule of expected completion times as well as the variability in these time, cost, and performance estimates. We can apply our equations to rank the desirability of projects in a proposed portfolio, thus aiding in the portfolio selection process. A stochastic extension to the Analytic Hierarchy Process (AHP) is also used in conjunction with our methodology to demonstrate the application of uncertainty calculations in managerial group choice situations.

Acknowledgements

The number of people that influenced this work is difficult to count. I have to thank all my professors and peers for their knowledge and points of view. I also need to thank my Ph.D. supervisor, Dr. Norman P. Archer, for his continued support and availability throughout my Ph.D. studies. I would also like to thank the Ph.D. secretary, Linda Kszyston, for her candor and availability in all matters related to my Ph.D. Last but not least, I would like to thank my family for their support, and to dedicate this thesis to the memory of my father, who died three years ago of lung cancer.

Table of Contents

Notation

$\aleph$ Adjacency matrix of a probabilistic network composed of nodes **N** = {1, 2, …, n} and activities **A** = {1, 2, …, m}.

t_{ij} Random duration of activity k in arc (i,j) where activity k belongs to path p ($k \in p$) described by the beta distributed density function $f_{ij}(t)$ with minimum, mean, and maximum parameters a_{ij}, μ_{ij}, and b_{ij}; correspondingly, range parameters $[a_{ij},b_{ij}]$ and shape parameters (α_{ij},β_{ij}).

T_i Random completion time at start node i with unknown probability density function $f_i(T)$.

T_j Random completion time at sink node j with unknown probability density function $f_j(T)$.

$\mathbf{R}_j$ Set of predecessor nodes directly connecting to node j.

T_{ij} Completion time of activity k in arc (i,j) if activity k is 100% critical.

T Completion time.

t Duration time.

$f_k(a_k,b_k,\alpha_k,\beta_k)$ Beta distributed duration time of activity k with range parameters $[a_k,b_k]$ and shape parameters (α_k,β_k).

λ_i Mean completion time at node i.

λ_j Mean completion time at node j.

$\underline{m}$ Minimum number of activities of a network of n nodes and m activities.

$\overline{m}$ Maximum number of activities of a network of n nodes and m activities.

$\overline{w}$ Maximum number of paths of a network of n nodes and m activities.

η Density coefficient of an AOA network with n nodes and m activities.

w Total number of paths p = 1, 2, …, w.

A_i Minimum completion time at node i.

A_j Minimum completion time at node j.

a_{ij} Minimum duration time of arc (i,j).

B_i Maximum completion time at node i.

B_j Maximum completion time at node j.

b_{ij} Maximum duration time of arc (i,j).

α_k First shape parameter of the beta distributed duration time of activity k.

β_k Second shape parameter of the beta distributed duration time of activity k.

μ_j Mean completion time at node j.

σ_j^2	Variance of the completion time at node j.
$\lvert z_{hi} \rvert$	Normalized criticality index of activity (h,i).
$\lvert z_{ij} \rvert$	Normalized criticality index of activity (i,j).
z_{hi}	Criticality index of activity (h,i).
z_{ij}	Criticality index of activity (i,j).
z_i	Criticality index at node i.
z_{in}	Criticality index of the arc (i,n)
RL	Run length.
ε	Error margin of the output variable.
$u_k=u_{ij}$	All-delayed direct cost of activity k in arc (i,j) of the maximum activity duration time (minimum cost).
$v_k=v_{ij}$	All-crashed direct cost of activity k in arc (i,j) of the minimum activity duration time (maximum cost).
$d_k=d_{ij}$	Direct cost of activity k in arc (i,j).
U	All-delayed (minimum) direct cost of the project.
V	All-crashed (maximum) direct cost of the project.
D	Total direct cost of the project.
I	Project indirect cost.
O	Minimum overhead (indirect) cost.
ΔO	Increment (slope) of the overhead (indirect) cost as a function of time.
C	Total cost.
C_a	Total cost of the minimum completion time.
C_b	Total cost of the maximum completion time.
C_{Max}	Maximum total cost.
C_{Min}	Minimum total cost.
U_j	Minimum cumulative direct cost occurring at maximum completion time (B_j).
V_j	Maximum cumulative direct cost occurring at minimum completion time (A_j).
U_{ij}	Minimum cumulative direct cost of activity k in arc (i,j).
V_{ij}	Maximum cumulative direct cost of activity k in arc (i,j).
D_j	Cumulative direct cost at node j.
I_j	Cumulative indirect cost at node j.
C_j	Cumulative total cost at node j.
A_{ij}	Minimum completion time of activity k in arc (i,j).
B_{ij}	Maximum completion time of activity k in arc (i,j).
λ_{ij}	Expected (average) completion time of activity k in arc (i,j).
μ_i	Expected (average) completion time of node i.
μ_{ij}	Expected (average) duration time of activity k in arc (i,j).
S_{ij}	Slack of activity k in arc (i,j).
γ_{ij}	Percentage of slack by which activity k in arc (i,j) has been delayed.
C_i	Total cost assigned to node i.
NPV	Net Present Value of the project.
H	Project planning horizon.

Q=IRR	Project's rate of return, equal to the Internal Rate of Return.
E	Income per time unit expected after project completion.
NFV	Net Future Value of the project.
s	Portfolio size.
$\mathbf{x}=[x_1,\ldots,x_s]$	Column vector indicating whether or not project k is included in the portfolio.
$\mathbf{q}=[q_1,\ldots q_s]$	Column vector indicating the performance of project k.
$\mathbf{t}=[t_1,\ldots,t_s]$	Column vector indicating the completion time of project k.
$\mathbf{c}=[c_1,\ldots,c_s]$	Column vector indicating the total cost of project k.
$\mathbf{r}=[r_1,\ldots,r_s]$	Column vector indicating the risk of project k.
$\Delta\mathbf{q}=[\Delta q_1,\ldots, \Delta q_s]$	Column vector indicating absolute variability of the performance of project k.
$\Delta\mathbf{t}=[\Delta t_1,\ldots, \Delta t_s]$	Column vector indicating absolute variability of the completion time of project k.
$\Delta\mathbf{c}=[\Delta t_1,\ldots, \Delta t_s]$	Column vector indicating absolute variability of the total cost of project k.
$\Delta\mathbf{r}=[\Delta r_1,\ldots \Delta r_s]$	Column vector indicating absolute variability of the risk of project k.
$\mathbf{x}^*=[x_1^*,\ldots,x_s^*\}$	Column vector indicating the optimal solution.
S_m	Set of mandatory projects.
P_j	Set of projects preceding project j.
M_j	Set of projects mutually exclusive with respect to project j.
w_t	Relative importance of time compared to cost and performance.
w_c	Relative importance of cost compared to time and performance.
w_q	Relative importance of performance compared to time and cost.
B	Total budget.
R	Risk preference of the group of decision makers.
g_k	Uncertainty to average ratio of project k.
$\bar{r}$	Average risk of the portfolio.
$\bar{t}$	Average time of the portfolio.
Z_k	Classification index of project k.
rq_k	Performance ratio for project k.
rt_k	Time ratio for project k.
rc_k	Cost ratio for project k.
ζ	Number of decision-makers.
ν	Number of criteria defined.
ω_i	Relative importance of criterion i.
$\mathbf{AA}$	Judgment matrix.
$\Delta\mathbf{AA}$	Uncertainties associated to the judgment matrix.
e_{ij}	Relationship (in relative importance) between criterion i and criterion j.
Δe_{ij}	Uncertainty of the relationship between criterion i and criterion j.
ω	Solution eigenvector.
$\mathbf{VV}_k$	Unnormalized eigenvector of weights.
$\Delta\mathbf{VV}_k$	Uncertainty of the unnormalized eigenvector of weights.

φ Parameter arbitrarily small for comparison purposes.

List of Figures

List of Tables

Chapter 1. Introduction

Project management is a managerial approach that integrates complex efforts by restructuring management and adopting special methods such as PERT/CPM, tradeoff analysis, and risk management in order to obtain better control and use of existing resources. Project management fosters cross-functional communication among operational islands across management and function gaps within the organization [63].

Driven to compete in global markets, organizations face considerable pressure to introduce new products with shorter life cycles satisfying minimum quality requirements at competitive prices. Business functions are merged in order to reduce the time it takes from concept to market. Management methodologies such as concurrent engineering, total quality management, and just in time manufacturing, among others, have been applied in order to cope with a fast paced, highly competitive, and dynamic global marketplace. In this context, comprehensive planning is a must. Successful project selection and management requires best practice, particularly in the case of knowledge and technology-based organizations in which successful R&D is a key ingredient [66].

According to Meredith and Mantel [58], the phases required for developing new products or updating existing ones are conceptual (preliminary design), definition (including detailed design), production (including prototype manufacturing), operations, and divestment. The traditional management approach to product development is sequential, with periodic revisions and iterations between phases. The concurrent engineering approach is to merge these phases in an on-going project evaluation and analysis process. Concurrent engineering [7][22][43][64][82] reduces time to market by squeezing the product development life cycle, carrying some of the product development phases and their tasks in parallel. A project consisting of the combination of two or more mutually inclusive tasks with pre-specified precedence relationships can in fact be considered a single project.

But what is a project? A project is an organized set of activities of finite duration to be accomplished, having a given purpose or goal (well-defined set of desired end results), with some unique elements and stakeholders (client, parent organization, project team, and the public). *A project is a combination of interrelated activities that must be executed in a pre-specified sequence in order to complete an entire task* [58].

Successful project management is the supervision of company resources, which involves project completion within the allocated time period, within the budgeted cost and at the proper specification level, resulting in positive benefits such as customer satisfaction among others. Time (indicated as a given schedule), cost (constrained by the budget), and performance (described as quality requirements for given specifications) are the three main project management dimensions [58].

Time, outlined as milestones or deadlines in a schedule, and cost, profiled by money allocations in a budget, are variables that should be minimized. Specifications are qualitative or quantitative descriptions of the deliverables as portrayed in the Statement of Work (SOW). The SOW is a list of the tasks or deliverables of the project organized as a hierarchy, where the key tasks are subdivided into a series of activities. The SOW allows decision makers to identify activity precedence.

These specifications can be of two types: a) specifications to be met, and b) specifications to be exceeded. Quality is a measure of conformance to specifications. For the first type of specification, quality is a function of specification variance: more (less) quality implies a lower (higher) variation from the specification given. For the second type of specification, quality is a function of the specification itself: more (less) quality implies exceeding (lagging) the specification given. Projects usually have two or more specifications to measure performance. Such specifications are project-specific. We consider the Net Present Value (NPV) and the Internal Rate of Return (IRR) to be our measures of performance.

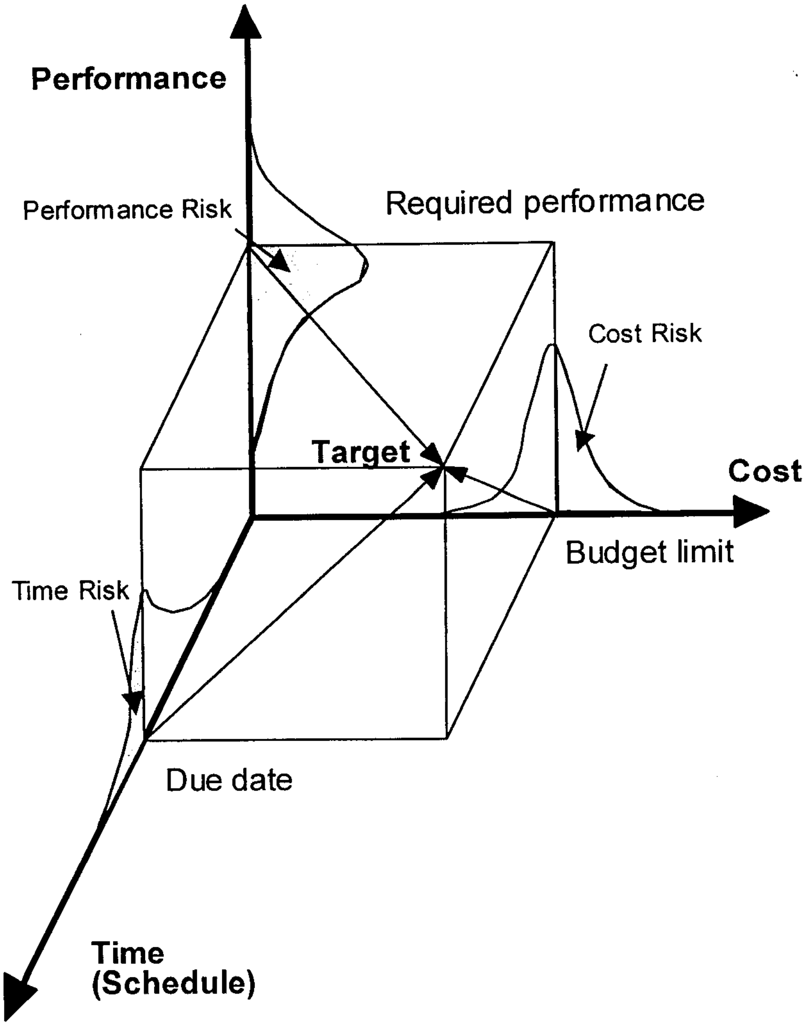


Figure 1. Time-Cost-Performance Tradeoff.
Adapted from Meredith and Mantel [58].

Time (schedule), cost (budget), and performance (quality or technical specifications) are the three project management (PM) prime objectives or targets (see Figure 1). Although the relationships among these dimensions vary from project to project, from time to time, and even within projects, it is possible to portray such dependencies as trade-offs. Klein [48] considers the uncertainties associated with each of these dimensions and portrays them as risk trade-offs. Figure 1 portrays the probabilistic nature of the project management dimensions by drawing a probability density function associated to each dimension. The due date is the time at which the project should be completed. The probability of not completing the project on schedule is the time risk. Also, the budget indicates the maximum cost allowed. The probability of having a cost greater or equal than the budget constitutes the cost risk. Performance is different. Assuming the performance measurement is a type of specification to be exceeded, performance risk is the probability of having a performance less than the performance requirement.

A great deal of good project management involves good project risk management. Project risk management can be defined as *the implications of the existence of significant uncertainty about the level of project achievement* [16]. Tight time, cost or performance targets increase time, cost or performance risks. A risk situation is often regarded as the existence of potentially very high and unacceptable costs or threats due to events assumed to be more or less likely to happen. This negative approach to risk leads to the idea that risk management essentially deals with removing or reducing the possibility of under-achievement. Risk analysis is not a 'throwing a dice' situation, but rather an area of study in which a pro-active, creative, and intelligent prior-planning approach is used, as opposed to entrenching in a defensive position [1][24][57][74][86].

Within this context, it is important to distinguish between risk and uncertainty. *Risk is the likelihood or probability of failure*, whereas *uncertainty is the variability of the relevant outcomes for a given risk or eventuality.* Brealey and Myers [11] define risk as to say that more things might happen (at present) than will happen (in the future). Uncertainty, on the other hand, is the degree in which an identified threat or risk (at present time after prior assessment) will (presumably, based on experience, historical data or assumptions) vary. Uncertainty is an identified (and quantified) risk. Still, the degree at which such identified risk will vary is unknown. Uncertainty thus constitutes the 'known' unknowns because although a specific risk has been identified, its actual impact is still unknown. Non-identified risks are 'unknown' unknowns because generally speaking a risk is non-quantified uncertainty about something not yet considered to be possible as a future outcome. We will assume throughout that risk identification has been successfully and thoroughly carried out and will focus on the risk due to the uncertainty for the most relevant variables previously identified by decision-makers. Risk sources are any factors that can affect the project's dimensions. Setting a tight time target such as an optimistic project deadline increases the project's time risk. Likewise, an unreasonably small budget increases cost risk and setting a minimum Net Present Value (NPV) increases performance risk. On the other hand, allowing slack times, contingency budget allocations or lowered NPV decreases time, cost, or performance risk, respectively [21][33][52][81].

The purpose of this thesis is to demonstrate some new methodologies for time, cost and performance tradeoffs in project management. Chapter two discusses the scheduling problem assuming unlimited resources, in order to obtain a probabilistic estimate of project completion for the time dimension. It considers the implications of project complexity in order to assess the practical usefulness of a model, based on beta distributed duration times, when applied to large projects. Chapter three explores the tradeoff between time and cost and proposes a model for optimal scheduling. By combining the findings presented in chapters two and three, decision makers are able to calculate the expected completion time at a given cost for a given performance, in other words, solve individual projects for the three project management dimensions. The next step is to solve for a set of projects (a project portfolio), which is called the portfolio selection problem. Chapter four proposes a model for portfolio selection that assists managers to priority order projects, using time and cost results calculated from individual projects and specifically considering risk tradeoffs. Chapter five discusses a modification of the Analytic Hierarchy Process (AHP), which is required to merge time-risk, cost-risk, and performance-risk for applications in portfolio selection. Finally, chapter six draws conclusions and proposes lines of further research.

Chapter 2. Project Completion Time

2.1 Introduction

One of the most important theoretical problems in project management is to obtain the distribution of the total project completion time in project networks [68]. The main approaches used are the Program Evaluation and Review Technique (PERT) and the Critical Path Method (CPM), both coincidentally developed in the same year (1959). PERT assumes three point estimates for probabilistic activity duration times in order to approximate project completion and the relative probability at each milestone, using the normal distribution [34]. CPM focuses on the criticality of each activity and the time-cost tradeoff in deterministic activity networks [78]. For practical and managerial purposes, what matters is the criticality of each activity within a PERT network, which can be assessed using a sound approach to calculate the completion time [90]. Critical activities are activities that, if delayed, would delay the entire project. A sequence of critical activities throughout the network is called a critical path. The critical path is the longest path in the network and it is possible to have more than one critical path at once. But unlike CPM, in stochastic activity networks the duration time of individual activities varies, so activities are critical for some combinations of duration times but may not be critical for other combinations. Therefore, activities have a given probability of being critical (i.e., being part of the longest path). We define for each activity the probability that the activity will be on the critical path as its criticality. The focus of this chapter is to describe an analytical method for calculating the theoretical distribution of the project completion time as well as the criticality of each activity.

Malcolm *et al.* [55] rely on the central limit theorem to postulate that the completion time can be portrayed using a normal distribution as a function of the cumulative mean and variance of all the activities within the longest path. Unfortunately, this results in unreliable (typically less than actual) completion times [26]. Martin [56] calculates the completion time by approximating task duration density functions using polynomials. Although accurate, Martin's method requires considerable calculation and is not easily suitable for software implementation. Kleindorfer [49] and Devroye [23] among others obtain lower bounds to the expected duration of the total project, based on node criticality, whereas Dodin and Elmaghraby [25] approximate such criticality indices. The latter is not entirely correct from a theoretical point of view, but the advantage of bounding the mean completion time from below is that closed form solutions can be obtained. Also, Dodin [27] tries to determine the k most critical paths as opposed to calculating completion times for each path. Monte Carlo simulation [83][85] is valid from a theoretical point of view, but it requires considerable calculation, which makes it impractical in the case of large networks. Keefer and Verdini [47] find a better way to estimate PERT activity time parameters [32][37].

We will show that the PERT assumption of a normally distributed project completion time typically leads project managers into optimistic planning, based on less than actual project completion estimates, due to a failure to consider the absolute bounds to project completion [28][38][54][73]. These bounds arise from the fact that the actual project completion time is the maximum sum of the duration of each and every path, which in turn is the result of adding the actual duration of its activities. It is common practice in PERT to estimate activity durations by using beta distributions [34]. Project completion cannot be an unbounded random variable because the sum of bounded (beta distributed) activity duration times yields bounded path (and project) completion times. The normal distribution cannot give upper and lower bounds on project completion times. PERT uses the same completion time algorithm as CPM, but applied to the mean. The problem is that this algorithm yields inaccurate results.

Also, the PERT textbook formula to calculate expected (mean) activity duration times, which are assumed to follow beta density functions, considers three parameters (minimum, most likely, and maximum), when in fact the beta distribution has four parameters: two range parameters and two shape parameters [54]. It turns out that the PERT formula used to calculate the mean as a function of the minimum (a), most likely (mode or m), and maximum (b) activity duration time estimates, (a+4m+b)/6, ignores how the biases to the right or left (related to the variance) affect the shape of the beta distribution.

2.2 PERT/CPM Networks

Network models can be used to schedule complex projects that consist of many activities. CPM can be used when the duration of each activity is known with certainty, to determine the duration of the entire project. It can also be used to determine how long activities in the project can be delayed without delaying the entire project. CPM was developed in the 1950s by researchers at du Pont and Sperry Rand [58]. If the duration of the activities is not known with certainty, PERT can be used to estimate the probability of the project being completed at any given deadline. PERT was developed in the late 1950s by consultants working on the development of the Polaris missile [15][31][62][88].

A project is a combination of interrelated tasks or activities that must be executed in some pre-specified sequence. Projects are described using probabilistic or deterministic activity networks, which are directed acyclic graphs. Let $\aleph$ denote the adjacency matrix of a probabilistic PERT/CPM network composed of nodes (vertices) $\mathbf{N} = \{1,2,\ldots,n\}$ and directed arcs $\mathbf{A} = \{(i,j) \mid$

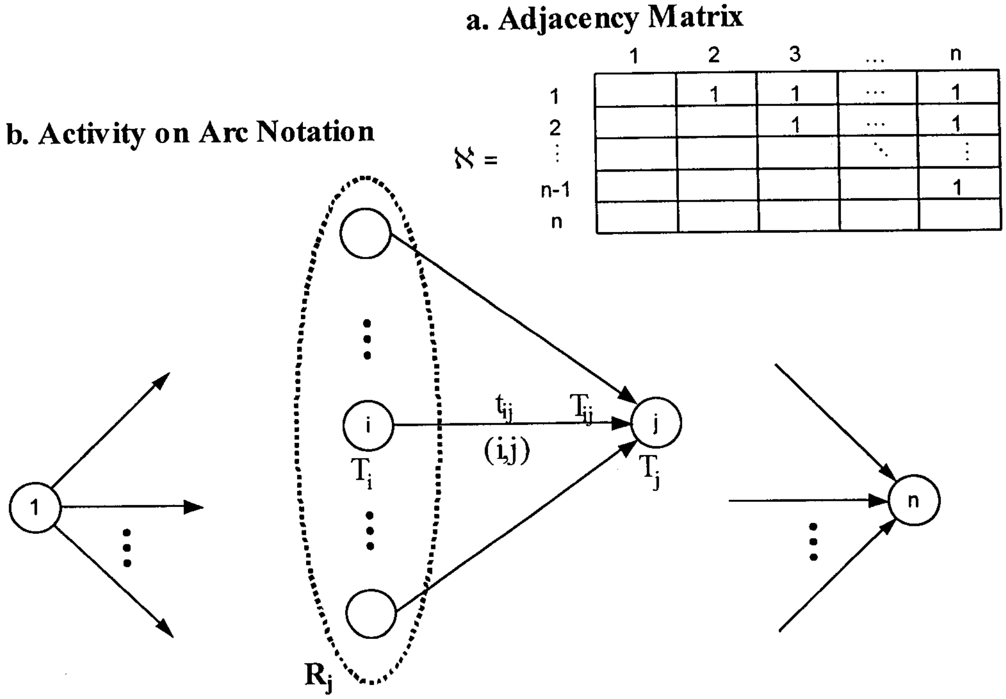
$i=1,\ldots,n-1,\ j=2,\ldots,n\}$ where n is the total number of nodes. Let m be the total number of activities so that the set of directed arcs, **A**, can also be denoted as **A** = $\{k \mid k=1,\ldots,m\}$. The duration of arc (i,j) is a random variable t_{ij} with known probability density function $f_{ij}(t)$ over the closed interval $[a_{ij},b_{ij}]$ where μ_{ij} denotes the mean (expected) duration of activity k in arc (i,j) and σ_{ij}^2 its variance. (Activity on Arc notation or AOA is implicit, where i indicates the node of origin and j the node of destination.) The completion time at sink node j, T_j, is the time at which all activities coming into j have been completed. The completion time at source node i, T_i, is the earliest time at which any activity k in arc (i,j) located between nodes i and j is allowed to start. (Notice that $T_i=T_j$ when i and j refer to the same node; i.e., i=j.) T_i (or T_j) is a random variable with unknown probability density function $f_i(T)$ (or $f_j(T)$). The purpose of our discussion is to describe how to accurately calculate the relevant probability density functions.

The adjacency matrix contains all precedence relationships. Figure 2a illustrates the adjacency matrix of a fully connected activity network. Row i indicates the node of origin while column j is the destination node for activity k at coordinates (i,j), explicitly specifying the position within the network for each activity. The nodes in directed acyclic networks are numbered in such way that an arc always leads from a smaller numbered node to a larger one. Let $\mathbf{R}_j$, j=2,…,n, denote the set of predecessor nodes connecting to node j. Let i be one such node ($i\in\mathbf{R}_j$). Figure 2b illustrates the notation. The completion time at node i is given by the random variable T_i, where T_i' is one random occurrence of T_i. If i is the only node in $\mathbf{R}_j$ (i.e., $|\mathbf{R}_j|=1$), then T_j' is given by the sum $T_i'+t_{ij}'$, where t_{ij}' is one random occurrence of t_{ij}. In general, when the number of arcs coming into node j is more than one (i.e., $|\mathbf{R}_j|>1$), the resulting completion time at j is the maximum completion time of all incoming arcs as indicated in equation (1). (Notice that

T is used to indicate completion time, whereas t indicates duration time; T includes the duration time of all preceding activities.)

$$T_j' = \underset{i \in R_j}{\mathrm{Max}}\{T_i' + t_{ij}'\}, j=2,\dots,n \qquad (1)$$

It is sometimes useful to denote activities using a single number k because it facilitates notation involving sets in which activity k is said to belong to path p for all p=1,...,w, where w is the total number of paths. (A path is a specific sequence of activities beginning at node 1 and ending at node n.) Conversely, denoting activities using their nodes of origin and destination facilitates writing equations for forward pass computations such as equation (1).



a. Adjacency Matrix

$$\aleph = \begin{array}{c|ccccc} & 1 & 2 & 3 & \dots & n \\ \hline 1 & & 1 & 1 & \cdots & 1 \\ 2 & & & 1 & \cdots & 1 \\ \vdots & & & & \ddots & \vdots \\ n\text{-}1 & & & & & 1 \\ n & & & & & \end{array}$$

b. Activity on Arc Notation

Figure 2. PERT/CPM Network.

Equation (1) is a stochastic sum across the network. The plus sign is used to denote the addition of the occurrence of two stochastic variables. $T_{ij}=T_i+t_{ij}$ indicates the completion time that would occur at node j if activity (i,j) happens to be critical across the network (longest duration time in a particular combination of random duration times), whereas t_{ij} is the duration time of activity k in arc (i,j). The completion time at node j is by definition the set of all maximum duration time combinations of the set $\mathbf{R}_j$ of all nodes i preceding node j.

Random duration times are described using probability density functions. In particular, PERT assumes that each activity duration time is given by a beta density function [58]. Range and shape parameters are required to specify beta density functions. The range parameters are a and b (minimum and maximum), and the shape parameters are α and β. Let f(x) be a beta density function as defined in equation (2). We chose a beta density function because it is commonly used in project management models to denote activity duration times [58].

$$f(x) = \frac{1}{\int_0^x t^{\alpha-1}(1-t)^{\beta-1}dt} \frac{(x-a)^{\alpha-1}(b-x)^{\beta-1}}{(b-a)^{\alpha+\beta-1}}, a < x < b \qquad (2)$$

The standardized beta density function varies between 0 and 1 (range parameters given) so that only the shape parameters are required. Range parameters are intuitively easy to understand and it is reasonable to expect decision-makers to use them and to provide their estimates. But shape parameters are difficult to grasp. So instead of specifying the shape parameters, decision-makers are asked to give the range and the most likely duration time (mode). From these the mean and variance are usually approximated in practice by (a+4m+b)/6 and $((b-a)/6)^2$, respectively. The problem is how to add the random variables of beta density functions across the activity network accurately in order to obtain a probability density function describing project completion time.

2.3 Stochastic Sum

Let the set p denote a path consisting of a sequence of n_p activities and let activity k be one of the activities in path p ($k \in p$). Also, let $f_k(a_k,b_k,\alpha_k,\beta_k)$ or simply $f_k(t)$ be a beta density function with range parameters a_k and b_k ($a_k<t<b_k$) and shape parameters α_k and β_k describing the duration of activity k, where $F_k(t)$ is the corresponding cumulative distribution. By definition (Hastings *et al.*, [41]), the mean of the beta distributed duration time for activity k is given according to equation (3).

$$\mu_k = \frac{a_k(\beta_k+1)+b_k(\alpha_k+1)}{\alpha_k+\beta_k+2} \tag{3}$$

The beta density function can be simplified to the standard beta distribution by assuming that the range parameters are 0 and 1. Let $t' = (t-a_k)/(b_k-a_k)$ be the standardized duration time ($0<t'<1$), where $a_k'=0$ and $b_k'=1$ denote the range parameters of the standard beta density function. Let μ_k' be the mean of the standardized beta distribution. Clearly, μ_k' is the relative distance between the original mean and the range parameters as indicated in equation (4).

$$\mu_k' = \frac{\mu_k - a_k}{b_k - a_k} \tag{4}$$

If we assume that the relationship between the shape parameters of the beta density function (α_k and β_k) and the shape parameters of the standardized beta density function (α_k' and β_k') is given by equations (5) and (6), we can obtain the mean of the standardized beta density function by substituting into equation (3) as indicated in equation (7).

$$\alpha_k' = \alpha_k + 1 \tag{5}$$

$$\beta_k' = \beta_k + 1 \tag{6}$$

$$\mu_k' = \frac{a_k'\beta_k' + b_k'\alpha_k'}{\alpha_k' + \beta_k'} = \frac{\alpha_k'}{\alpha_k' + \beta_k'} \tag{7}$$

Rearranging the terms of equation (7) yields equation (8).

$$\left(\frac{\beta_k'}{1}\right)\left(\frac{1}{\alpha_k'}\right) = \left(\frac{1}{\mu_k'}\right)\left(\frac{1-\mu_k'}{1}\right) \tag{8}$$

Figure 3 shows all three types of standard beta distributions for different combinations of shape parameters. U-shaped beta distributions occur when the sum of the shape parameters is less than 2. J-shaped beta distributions occur when the sum of the shape parameters is greater than or equal to 2 and less or equal than 3. Bell-shaped beta distributions occur when the sum of the shape parameters is greater than 3.

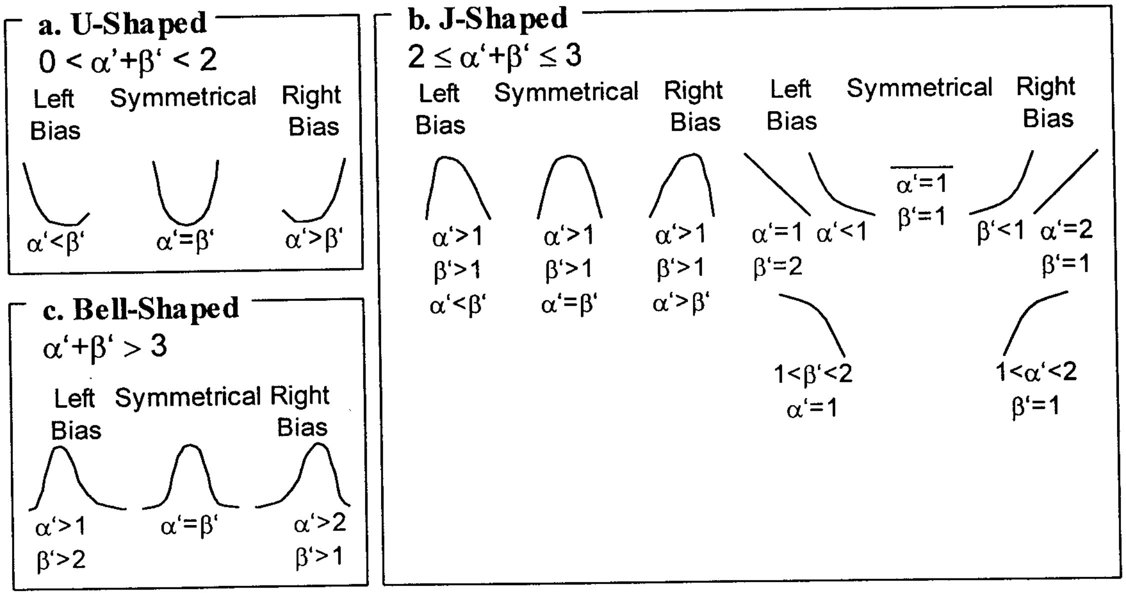


Figure 3. Shapes of the Beta Distribution.

It is common practice to portray activity duration times using bell-shaped beta distributions [55]. Only one interpretation of equation (8) provides the simplest system of two equations portraying α' and β' as a function of μ' that guarantees a bell-shaped beta density function (as opposed to U-shaped or J-shaped) for any given value of μ'. The four alternative systems of equations (cases) for β' and 1/α' consistent with equation (8) are:

i. $\beta'=\left(\frac{1-\mu'}{\mu'}\right)^{\frac{x}{y}}$ and $\frac{1}{\alpha'}=\left(\frac{1-\mu'}{\mu'}\right)^{\frac{y-x}{y}}$ so that x/y+(y-x)/y=1.

ii. $\beta'=1-\mu'$ and $\frac{1}{\alpha'}=\frac{1}{\mu'}$.

iii. $\beta'=1$ and $\frac{1}{\alpha'}=\frac{1-\mu'}{\mu'}$ if μ'>0.5 (right bias) or $\alpha'=1$ and $\beta'=\frac{1-\mu'}{\mu'}$ if μ'<0.5 (left bias).

iv. $\beta'=\frac{1}{\mu'}$ and $\frac{1}{\alpha'}=\frac{1-\mu'}{1}$.

There are an infinite number of combinations of x and y for the first case, leading to a system of equations for α' and β' consistent with equation (8). According to the scientific precept known as Ockham's razor, attributed to the English philosopher William of Ockham [60], all things being equal, the simplest explanation tends to be the truth. Clearly, a case in which there are infinite possibilities is not the simplest case, so alternative i) should be discarded. The second alternative is also discarded. It leads to U-shaped beta distributions because 0<μ'<1 so that α'= μ' and β'=1-μ' must be between 0 and 1 as well (see Figure 3a). The third alternative is also rejected because it corresponds to J-shaped beta distributions since in that case, either α' or β' equals 1 (see Figure 3b). The last alternative is the only one that ensures a bell-shaped beta distribution in which the sum of α' and β' is greater than 3 (see Figure 3c). The minimum value for α'+β' occurs when μ'=0.5 so that α'=β'=1/0.5=2 and α'+β'=4. All other values for μ' lead to values of α'+β' greater than 4. Consequently, the shape parameters of the standardized beta density function describing the standardized duration time of activity k are a function of the standardized mean as indicated in iv). Equations (9) and (10) portray the results from case iv).

$$\alpha_k' = \frac{1}{1-\mu_k'} \tag{9}$$

$$\beta_k' = \frac{1}{\mu_k'} \tag{10}$$

Substituting μ_k' from equation (4) into equations (9) and (10) yields equations (11) and (12).

$$\alpha_k = \frac{b_k - a_k}{b_k - \mu_k} \tag{11}$$

$$\beta_k = \frac{b_k - a_k}{\mu_k - a_k} \tag{12}$$

The standardized variance is by definition (Hastings *et al.*, [41]) given according to equation (13).

$$\sigma_k'^2 = \frac{\alpha_k'\beta_k'}{(\alpha_k'+\beta_k')^2(\alpha_k'+\beta_k'+1)} \tag{13}$$

Also, the variance is $(b-a)^2$ times the standardized variance as indicated in equation (14).

$$\sigma_k^2 = (b_k - a_k)^2 \sigma_k'^2 \tag{14}$$

It is known that the mean and variance of a random variable that is the result of adding a sequence of independent random variables is the sum of the mean and variances of each random variable added. Therefore, assuming independence among activity duration times, the mean and variance of the duration time of path p is given by the sum of the mean and variance of its activities as indicated in equations (15) and (16).

$$\mu_p = \sum_{k \in p} \mu_k \tag{15}$$

$$\sigma_p{}^2 = \sum_{k \in p} \sigma_k{}^2 \tag{16}$$

Also, the minimum and maximum path duration times are given by the sum of the minimum and maximum of the activities in the path according to equations (17) and (18).

$$a_p = \sum_{k \in p} a_k \tag{17}$$

$$b_p = \sum_{k \in p} b_k \tag{18}$$

The shape parameters of the path can be obtained by applying equations (11) and (12) if the duration time of the path is approximated by a beta density function with parameters a_p, b_p, α_p, and β_p.

The stochastic sum can be used to add node completion times and activity duration times. Let $f_{ij}(t)$ be a beta density function with range parameters a_{ij} and b_{ij} and shape parameters α_{ij} and β_{ij}, describing the duration of activity (i,j) and $F_{ij}(t)$ be the respective cumulative distribution for activity (i,j). Also, let $f_i(T)$ be assumed to be the beta density function describing the completion time of node i, where $F_i(T)$ is its cumulative distribution. Let the function $f_{ij}(T)$ denote the stochastic sum between node i and activity (i,j) as indicated in equation (19).

$$f_{ij}(T) = f_i(T) \oplus f_{ij}(t) \tag{19}$$

The parameters of the function $f_{ij}(T)$ are obtained from the parameters of the functions $f_i(T)$ and $f_{ij}(t)$ according to equations (9) to (18). $F_{ij}(T)$ is the cumulative distribution of the stochastic sum between node i and activity (i,j), calculated by integrating $f_{ij}(T)$, as shown in equation (20).

$$F_{ij}(T) = \int_0^T f_{ij}(T)dT \tag{20}$$

2.4 PERT Completion Time

The coefficients of the function $f_{ij}(T)$ are essentially obtained by adding mean duration times and minimum and maximum times of the preceding activities and adjusting for the variance. PERT involves the addition of mean duration times. But activity networks are a combination of entangled paths and not a single path. The concept of stochastic sum applies only to specific paths. The problem is how to calculate the completion time at nodes with several incoming activities. It is tempting to extend equation (1) and apply it to mean completion times. In fact, that is exactly what PERT is all about. PERT assumes the mean completion time at node j is the maximum of the mean completion times of all the arcs preceding node j [28]. Let λ_j denote the mean completion time at node j for all j=1,...,n, where λ_1=0. Then, the mean completion time at node j in PERT is given according to equation (21).

$$\lambda_j = \operatorname*{Max}_{i\in R_j}\{\lambda_i + \mu_{ij}\} \ \forall\, j=2,\ldots,n \tag{21}$$

As we have seen in the previous section, adding the mean completion time of each node i and the corresponding activity in arc (i,j) is statistically acceptable because both means are in sequence and the result would be the mean of activity (i,j) if the activity is critical. But assuming that the mean at node j is the maximum of these is not accurate. This is because we do not know a priori which activity is critical. It may very well be that several activities are critical in different degrees (with different probabilities) for different duration time combinations. Besides, equation (1) applies to random variables and not just to expected values. To illustrate, consider two activities discretely distributed but arranged in parallel. Assume that the first activity can have a duration time of 5 or 8 with equal probability (t_1={5,8}), whereas the duration time of the second activity can be 6 or 7 with equal probability (t_2={6,7}). PERT would calculate the mean duration time of the first activity, (5+8)/2=6.5, and the mean duration time of the second activity, (6+7)/2=6.5, and assume the mean completion time of both activities to be μ_{PERT}=Max(μ_1,μ_2)=Max(6.5,6.5)=6.5. But in fact, duration times are random variables, which means that there are four possible combinations for Max(t_1,t_2) indicating project completion: Max(5,6)=6, Max(5,7)=7, Max(8,6)=8, and Max(8,7)=8. The mean completion time is in fact the average of these: $\mu_{THEORETICAL}$=(6+7+8+8)/4=7.25. In this case PERT underestimates the completion time because it does not consider the probability distributions, which describe the random behavior of activity duration times.

So how can we accurately estimate expected project completion? One way is to consider all path combinations, calculate the duration time of each path by adding the duration time of its activities, and then obtain the joint probability density function of these and calculate its mean. Unfortunately, the number of paths grows exponentially as the number of nodes increases. In other words, the computational effort increases as the complexity of the network increases.

2.5 Project Complexity

The minimum number of activities, $\underline{m}$, for a serial network of n nodes is n-1. The maximum number of activities, $\overline{m}$, for a fully connected network of n nodes is n(n-1)/2 [51]. Clearly, the minimum number of paths for an all-serial network, $\underline{w}$, is one. What is the maximum number of paths? The maximum number of paths occurs in fully connected networks of n nodes and n(n-1)/2 activities. All paths must include the first and last nodes. The total number of combinations (subsets) of all intermediate nodes gives the total number of paths in fully connected networks. In a fully connected network of n nodes, there are n-2 intermediate nodes (n nodes minus the first and last nodes), so that the maximum number of paths, $\overline{w}$, is the total number of subsets of a set of size n-2. According to the binomial theorem [61], this is given by equation (22).

$$\overline{w} = 1 + \binom{n-2}{1} + \binom{n-2}{2} + \ldots + \binom{n-2}{n-2} = 2^{n-2} \tag{22}$$

Let η be defined as the density coefficient for an AOA network with n nodes and m activities, indicating how nearly all-serial or all-parallel (fully connected) the network is, as shown in equation (23). The density coefficient is the proportional distance between the actual and the minimum compared to the maximum number of activities. (Notice, for m=$\underline{m}$, η=0, and for m=$\overline{m}$, η=1.)

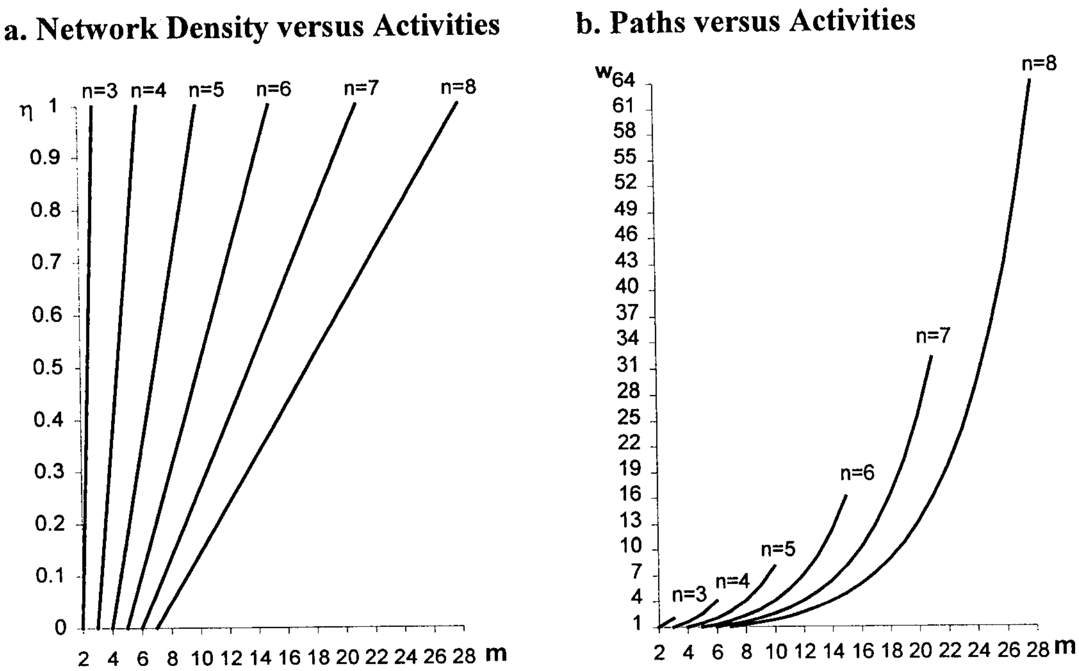
$$\eta = \frac{m - \underline{m}}{\overline{m} - \underline{m}} = \frac{2(m-n+1)}{(n-1)(n-2)}, \; n \geq 3 \tag{23}$$

The total number of paths, w, is then defined as a function of both n and η according to equation (24), where $\lfloor x \rfloor$ is the floor function (truncation of the fractional component) of x.

$$w \approx \left\lfloor 2^{\eta(n-2)} \right\rfloor \tag{24}$$

When η=0 for minimally connected (all-serial) networks, w=$\underline{w}$=2^0=1 and when η=1 for fully connected (all-parallel) networks, w=$\overline{w}$=2^{n-2}. Equation (24) portrays exponential growth mediated by η. The network density coefficient from equation (23) measures network complexity and equation (24) provides the number of paths for a given combination of activities and nodes.

Figure 4a plots network density (η on the vertical axis) as a function of the number of activities (m on the horizontal axis) and network size (n as different lines in the graph) according to equation (23). The relationship between the number of activities (m) and network density (η) is linear. It is clear from Figure 4a and equation (23) that the rate (given by the slope), at which network density (η) grows, decreases as network size (n) increases.


a. Network Density versus Activities

b. Paths versus Activities

Figure 4. Network Density (η) and Complexity (w).

Figure 4b plots the number of network paths (w on the vertical axis) as a function of the number of activities and network density (m which determines η along the horizontal axis) for different network sizes according to equation (24). Although the number of paths (w) increases exponentially as network size (n) and network density (η) increase, the rate at which such exponential growth occurs (η) decreases as network size (n) increases.

The total number of paths is important because more paths increase the computational effort required to estimate the expected project completion time and approximate the probability distribution of the network completion time. Figure 4 shows that although the computational effort tends to increase exponentially as network size increases (exponential time problem), the range for the number of activities $(\overline{m}-\underline{m})$ increases for larger networks. Consequently, the probability of having complex networks and significant exponential growth in the number of paths decreases as the number of nodes increases. The total number of paths is further constrained by the variability within the network of activity duration times and the resulting range and variability in pathway duration times (absolute bounds to completion time). We shall demonstrate that when all duration times are assumed to follow a beta distribution, it is possible to calculate completion time in a polynomial number of steps (polynomial time).

2.6 Absolute Bounds

PERT assumes a beta density function based on three-point estimates to portray activity duration times. We also assume beta distributed activity duration times, but unlike PERT, we do not assume project completion time to be normally distributed. Using a normal distribution implicitly considers project completion to be unbounded. We assume a beta distribution of completion time at all the intermediate and final nodes. But the beta distribution has a minimum and a maximum, and since the completion time at each node j is the stochastic sum of beta distributions with minimum and maximum values, there must be a minimum and a maximum at each node j for all $j=2,\ldots,n$.

Consider a set of paths arranged in parallel with beta distributed duration times, each with a different minimum and maximum. Equation (1) indicates that the resulting completion time is the maximum of all these randomly distributed path duration times. What is the minimum completion time possible? The minimum completion time must be the maximum of the minimum completion time of each and every path. The same reasoning applies to the analysis of maximum path duration times: the maximum completion time must be the maximum of the maximum completion time of each and every path. Therefore, although equation (1) cannot be applied to include the mean and the variance of path duration times, it can be applied to the range (minimum and maximum) in which the randomly distributed completion time is allowed to vary. Let A_j and B_j be the minimum and maximum completion time at node j for all j=2,...,n where $A_1=0$ and $B_1=0$ (by definition, the first node does not indicate completion time). Then, the minimum and maximum completion times at each node j are given according to equations (25) and (26).

$$A_j = \underset{i \in R_j}{\mathrm{Max}}(A_i + a_{ij}), \text{j=2,...,n} \tag{25}$$

$$B_j = \underset{i \in R_j}{\mathrm{Max}}(B_i + b_{ij}), \text{j=2,...,n} \tag{26}$$

These are absolute bounds to the completion time at node j because no T_j can be less than A_j nor greater than B_j at node j, as shown in equation (27).

$$A_j \leq T_j \leq B_j \tag{27}$$

2.7 Joint Integration

Equations (25) and (26) provide the range parameters of the project completion time (a and b) when j=n ($a=A_n$, and $b=B_n$). The shape parameters can be obtained as a function of the minimum, maximum and mean according to equations (11) and (12). But calculating the mean (and by extension the variance) requires knowing the probability density function of the completion time at each node. Denote the probability density function of the completion time at node j by $f_j(T)$, where $F_j(T)$ is the corresponding cumulative distribution. Assuming that all the stochastic completion times between each node i and each activity (i,j) given by the stochastic sum $f_{ij}(T)=f_i(T)\oplus f_{ij}(t)$ as indicated in equation (17) for all nodes i preceding node j ($i\in\mathbf{R}_j$) are independently distributed, the joint cumulative distribution at node j, $F_j(T)$, is given by the product of all $F_{ij}(T)$ as shown in equation (28). (Notice that $F_{ij}(T)$ is the cumulative distribution of $f_{ij}(T)$.)

$$F_j(T)=\prod_{i\in R_j}F_{ij}(T) \tag{28}$$

Then, the probability density function of the completion time at node j, $f_j(T)$, is given by the derivative of the cumulative distribution at node j as indicated in equation (29).

$$f_j(T)=\frac{\partial F_j(T)}{\partial T} \tag{29}$$

The mean and the variance of the completion time at each node j, μ_j and σ_j^2, are given according to equations (30) and (31) for all j=2,...,n.

$$\mu_j=\int_{A_j}^{B_j}Tf_j(T)dT \tag{30}$$

$$\sigma_j^2=\int_{A_j}^{B_j}(T-\mu_j)^2 f_j(T)dT \tag{31}$$

Assuming that f_j is a beta distribution, equations (11) and (12) are then used to calculate the shape parameters of the beta distributed completion time at node j, α_j and β_j. The probability density function of the project completion time is a beta distribution with range parameters $a=A_n$ and $b=B_n$ and shape parameters $\alpha=\alpha_n$ and $\beta=\beta_n$. In practice, the actual integration is done numerically.

By regarding activity networks as a combination of all-serial and all-parallel sub-networks, the maximum number of paths that need to be considered in order to obtain the beta density function describing project completion time is exactly the same as the number of activities in the network. This is because only arcs (activities) require computational effort in order to incorporate such duration time into the node's completion time. Solving for m from equation (23) yields equation (32).

$$w = m = \frac{(n-1)(\eta(n-2)+2)}{2} = \frac{\eta n^2}{2} + n\left(1 - \frac{3\eta}{2}\right) + \eta - 1 \qquad (32)$$

Equation (32) is a quadratic polynomial in n. Hence, calculating probabilistic project completion is a problem with polynomial complexity (requiring a polynomial number of steps as a function of problem size to solve the problem), not exponential complexity (requiring an exponential number of steps as a function of problem size to solve the problem).

2.8 Criticality Index

Criticality is the probability for any given activity to be in the longest (critical) path; in other words, it is the percentage of times in which the activity was in the longest path for all the random occurrences of duration times. Criticality is important because delays in critical activities are very likely to delay the entire project. Therefore, criticality is used for managerial purposes to keep under control the duration of all the critical activities.

Let h≠i denote two nodes preceding node j, so that arcs (h,j) and (i,j) denote two parallel activities coming into node j. The completion time of activities (h,j) and (i,j) is given by the probability density functions $f_{hj}(T) = f_h(T) \oplus f_{hj}(t)$ and $f_{ij}(T) = f_i(T) \oplus f_{ij}(t)$; $F_{hj}(T)$ and $F_{ij}(T)$ denote the respective cumulative distributions. At any given completion time T, the probability of activity (i,j) to have a completion time between T and T+dT is $F_{ij}(T+dT)-F_{ij}(T)$, which equals $f_{ij}(T)dT$ for infinitesimally small dT. Also, the probability of activity (h,j) to have a completion time less than T is $F_{hj}(T)$, and the probability of all activities (h,j) where $h\neq i$ to have a completion time less than T is given by the product $\Pi_{h\neq i}F_{hj}(T)$. Let $|z_{ij}|$ denote the normalized criticality index of activity (i,j), which is the probability (normalized) of activity (i,j) to be the longest at node j. Then, $|z_{ij}|$ is given according to equation (33).

$$|z_{ij}| = \int_{A_j}^{B_j} f_{ij}(T)\prod_{h\neq i} F_{hj}(T)dT = \int_{A_j}^{B_j} g_{ij}(T)dT, \qquad (33)$$

$$\text{where } g_{ij}(T) = \frac{\partial F_j(T)}{\partial T_{ij}} = f_{ij}(T)\prod_{h\neq i} F_{hj}(T) \text{ and } F_j(T) = \prod_{i<j} F_{ij}(T)$$

The cumulative distribution of the completion time at node j, $F_j(T)$, is calculated in equation (28) as the product of the cumulative distribution of the completion time of each and every activity coming into node j, $F_{ij}(T)$, where $F_{ij}(T)$ is the cumulative distribution of the probability density function obtained by adding the completion time at node i, and the duration time of activity (i,j): $f_{ij}(T) = f_i(T) \oplus f_{ij}(t)$. T_{ij} is a random variable described by the beta distributed probability density function denoted as $f_{ij}(T)$ indicating the completion time of activity (i,j), so that the function $g_{ij}(T)$ is the partial derivative of the cumulative distribution of the completion time at node j with respect to T_{ij}.

Although the function $|z_{ij}|$ indicates the probability of activity (i,j) to be longer than all other activities (h,j) for $h\neq i$, that is not the criticality index of activity (i,j), because the criticality index is the probability of being in the longest path, which includes all activities and not just the activities immediately preceding node j. Since $|z_{ij}|$ is calculated by integrating the partial derivative of $F_j(T)$ with respect to T_{ij}, the sum of all $|z_{ij}|$ for i<j equals one because that would in fact include the integral of all partial derivatives, as shown in equation (34a).

$$\sum_{i<j} |z_{ij}| = 1 \tag{34a}$$

Substituting h for i and i for j from equation (34a) yields equation (34b).

$$\sum_{h<i} |z_{hi}| = 1 \tag{34b}$$

The function $|z_{ij}|$ can be used in combination with the properties of the criticality index of activities and nodes in stochastic networks to calculate the criticality index of each activity using backward pass calculations. Backward pass calculations are node and arc computations in which the higher numbered nodes are computed first and computations proceed from the last to the first node. Let z_{hi} and z_{ij} be the criticality indices of activities (h,i) and (i,j). Also, let z_i and z_j denote the criticality indices of nodes i and j, respectively. Figure 5 illustrates the notation.

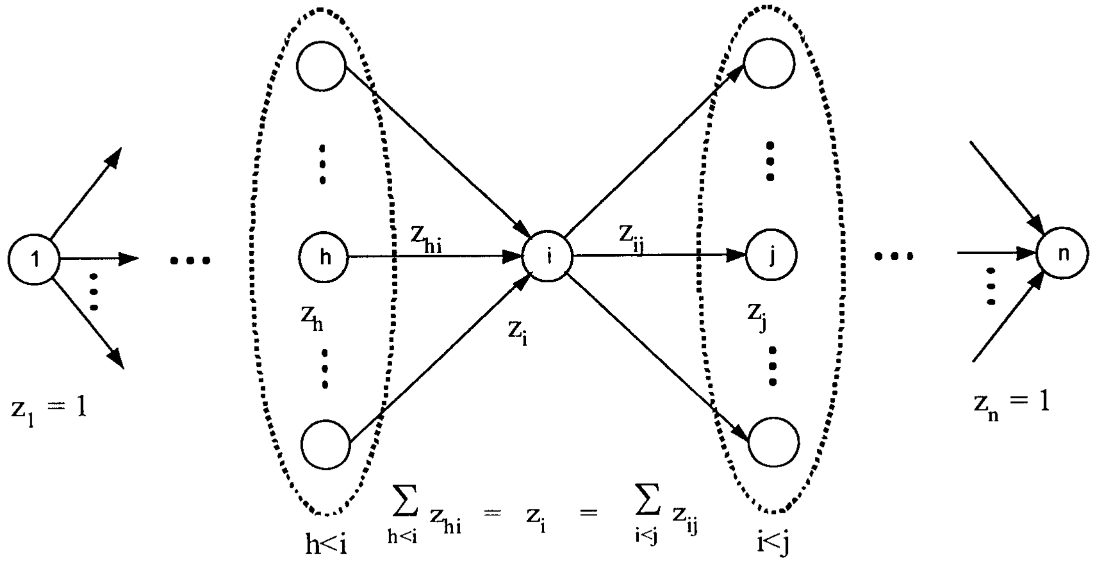


Figure 5. Criticality across Activity Networks.

First, the criticality index of the first and last node equals one, because these nodes are always in the longest path. Hence, z_1=1 and z_n=1. Second, for any node i, the sum of the criticality indices of the activities coming into node i equals the sum of the criticality indices of the activities coming out of node i as shown in equation (35), because in all cases in which node i is critical, there must be one activity (h,i) for h<i critical and one activity (i,j) for i<j critical corresponding to the critical sub-path h→i→j, such that h<i<j.

$$\sum_{h<i} z_{hi} = \sum_{i<j} z_{ij} \tag{35}$$

Such a sum corresponds to the criticality index of node i ($0<z_i<1$), as shown in equation (36).

$$z_i = \sum_{h<i} z_{hi} \tag{36}$$

Dividing equation (36) by z_i yields equation (37).

$$\frac{z_i}{z_i} = \frac{\sum_{h<i} z_{hi}}{z_i} = 1 \tag{37}$$

Equating equations (34b) and (37) yields equation (38).

$$\sum_{h<i} \left| z_{hi} \right| = \frac{\sum_{h<i} z_{hi}}{z_i} \tag{38}$$

It follows that the normalized criticality index of activity (h,i) equals the criticality index of activity (h,i) divided by the criticality index of node i as shown in equation (39).

$$\left| z_{hi} \right| = \frac{z_{hi}}{z_i} \tag{39}$$

Since the criticality index of the last node equals one, z_n=1, from equation (39) we have that the criticality index of all the activities (i,n) for i<n coming into the last node (n) equals their respective normalized criticality index as shown in equation (40).

$$z_{in} = |z_{in}| \tag{40}$$

Also, from equations (35) and (36), we have that z_i can be calculated as the sum of all the activities (i,j) for i<j coming out of node i as portrayed by equation (41).

$$z_i = \sum_{i<j} z_{ij}\,,\ i=n-2,\ldots,1,\ j=n-1,\ldots,2 \tag{41}$$

Finally, solving for z_{hi} from equation (39) yields equation (42). Equation (42) can be applied recursively from the last node to the first node in order to calculate the criticality index of each activity in the network through numerical integration.

$$z_{hi} = |z_{hi}|z_i = |z_{hi}|\sum_{i<j} z_{ij}\,,\ i=n-2,\ldots,1,\ j=n-1,\ldots,2 \tag{42}$$

2.9 Simulation

Simulation is the process of representing reality (by copying or imitating its behavior) as close as practically possible. Simulation usually involves a mathematical-logical model developed using the computer for experimentation and testing [29]. Simulation is important in the present work because it is used to test the accuracy of our calculation procedure.

Monte Carlo simulation is one of the most popular forms of computer-based simulation. Halton [39] defines the Monte Carlo method as "representing the solution of a problem as a parameter of an hypothetical population, and using a random sequence of numbers to construct a sample of the population, from which statistical estimates of the parameter can be obtained".

In Monte Carlo simulation, a set of variables (some or all of them random) is used as input. A mathematical relationship (expressed in a set of equations) links the input variable(s). By successively generating random variables according to the random behavior they follow, alternative values for the output variable(s) are obtained [40][50][76][92].

When the system is numerically modeled using Monte Carlo simulation, RL alternative results (outputs) are obtained. (RL stands for Run Length). Nevertheless, only one result (estimate) is required. Such result is always an approximation to the theoretical value (μ). Figure 6 shows the apparent random behavior around the theoretical value when each output is plotted against the iteration in which it was generated. The output (all RL observations) can be analyzed using a histogram to find the best fit for the data from available probability density functions (pdf's), as long as the system is in a steady-state process. Statistical measures such as mean and standard deviation are usually obtained instead of the histogram when the estimate and its closeness to the theoretical value are more important than the nature of the distribution itself.

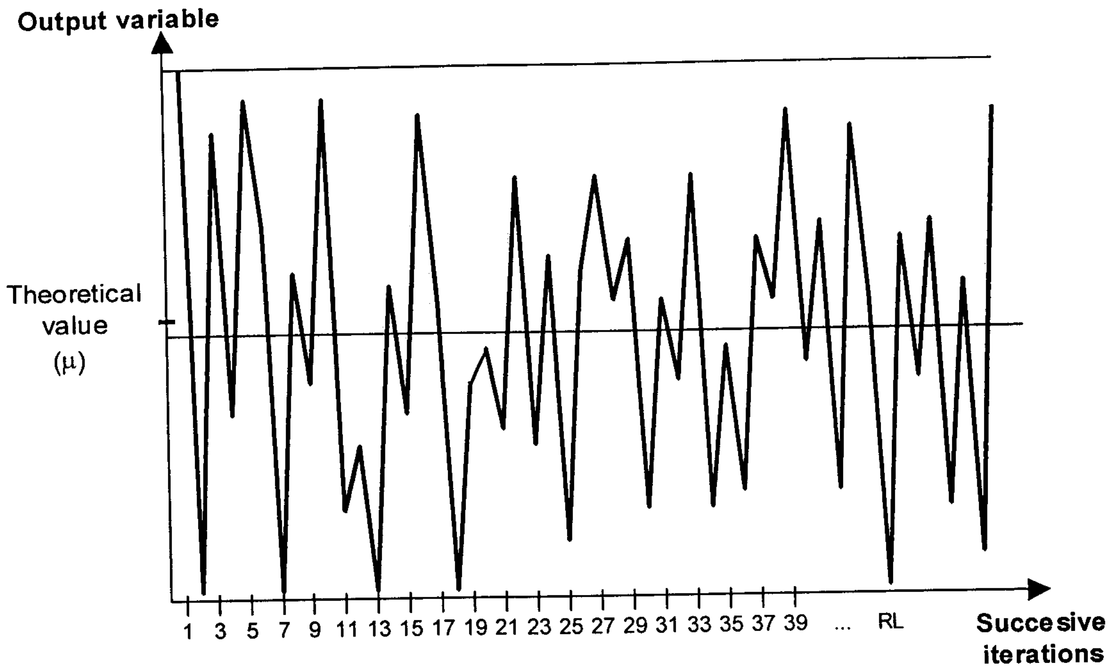


Figure 6. Fluctuation Around the Theoretical Value of the Simulation Process.

Figure 7 shows the behavior of the output variable when the cumulative average (the average of all available observations) is obtained for each estimate. As more observations become available (larger RL), the random noise due to the uncertainty of the output variable is reduced and the average tends to get closer and closer to the theoretical value. For the purposes of our discussion, all input variables are always properly modeled (activities are assumed to follow a beta density function and random numbers following such distribution are generated for each iteration). Since the system is stable from the start, no warm-up period is required.

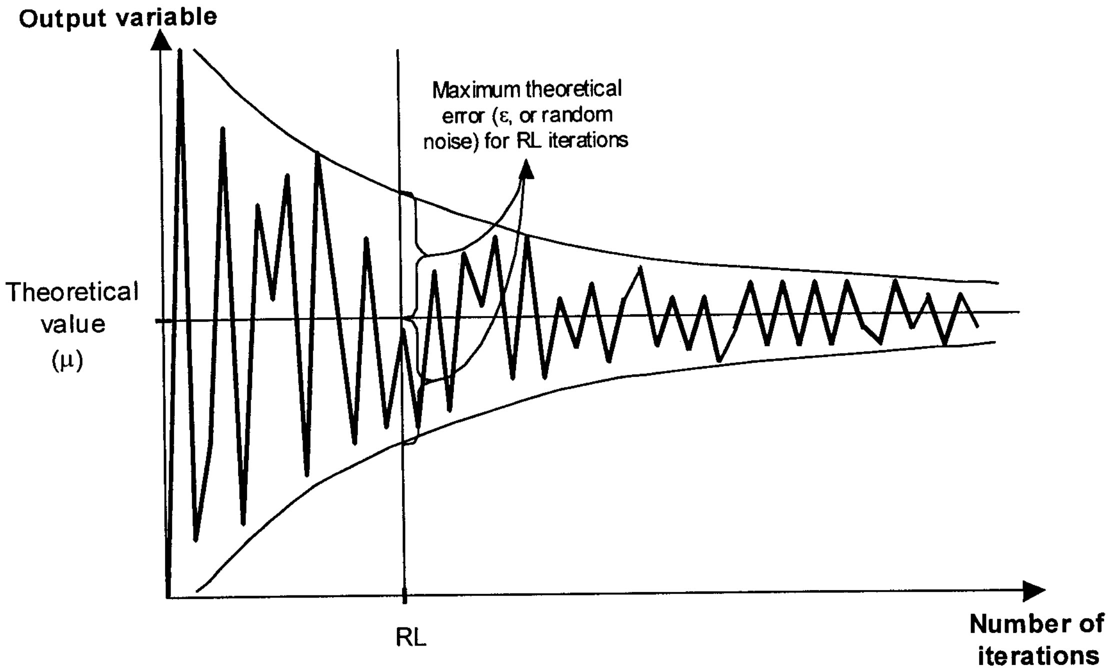

Output variable
Maximum theoretical error (ε, or random noise) for RL iterations
Theoretical value (μ)
RL
Number of iterations

Figure 7. Trend Towards the Theoretical Value when the Cumulative Average (Mean) is Calculated, as More Observations (Iterations) are Available.

The larger the value of RL the smaller the random noise inherent to the simulation process is. Nevertheless, a larger run length requires more time consuming computations. The trick is to find a value for RL large enough to estimate the output variable within a reasonable error margin (ε). In practice, this is done by successive analysis of the standard deviations obtained when new iterations are included [59]. Exponential smoothing can be used to determine the stopping point. The details are beyond the scope of our work and are not included as part of the discussion. Finally, it is important to mention that there are software packages such as @Risk[1] by Palisade Software Corporation that implement simulation estimates for project networks.

[1] http://www.palisade.com/

2.10 Example

Consider the example shown in Figure 8. Nodes are numbered (in italics) from 1 to 7 and activities are numbered (in bold) from 1 to 9. Each activity has a minimum, average or mean, and maximum duration time. In PERT (Figure 8a), mean duration times are recursively added. The mean duration time at node 1 is λ_1=0, at node 2 is the mean at node 1 plus the mean between nodes 1 and 2 corresponding to activity (i,j)=(*1*,*2*) or k=**1**, which is λ_2=0+11=11. A similar reasoning applies to node 3. For node 4, the mean duration time is assumed to be the maximum of the mean at node 2 plus the mean of the activity between node 2 and node 4 ($\lambda_2+\mu_{24}$) and the mean at node 3 plus the mean of the activity between nodes 3 and 4 ($\lambda_3+\mu_{34}$). The PERT estimate for the average project completion time at node 7 is 61, with a variance of 6.30. These results are plotted as a normal density function in Figure 9.

Figure 8b shows how to solve the problem according to equations (11) and (12) and equations (25) to (31) by brute force (6 steps in this particular case) by considering all possible paths. There are 6 alternative paths. The beta distribution describing the duration time of each of these paths is obtained by stochastic sum according to equations (11) and (12) and (25) to (31). Finally, the beta distribution describing project completion at node 7 is calculated by joint numerical integration according to equations (25) to (31). The mean and variance of the project completion time obtained in this way are 62.94 and 2.65, respectively.

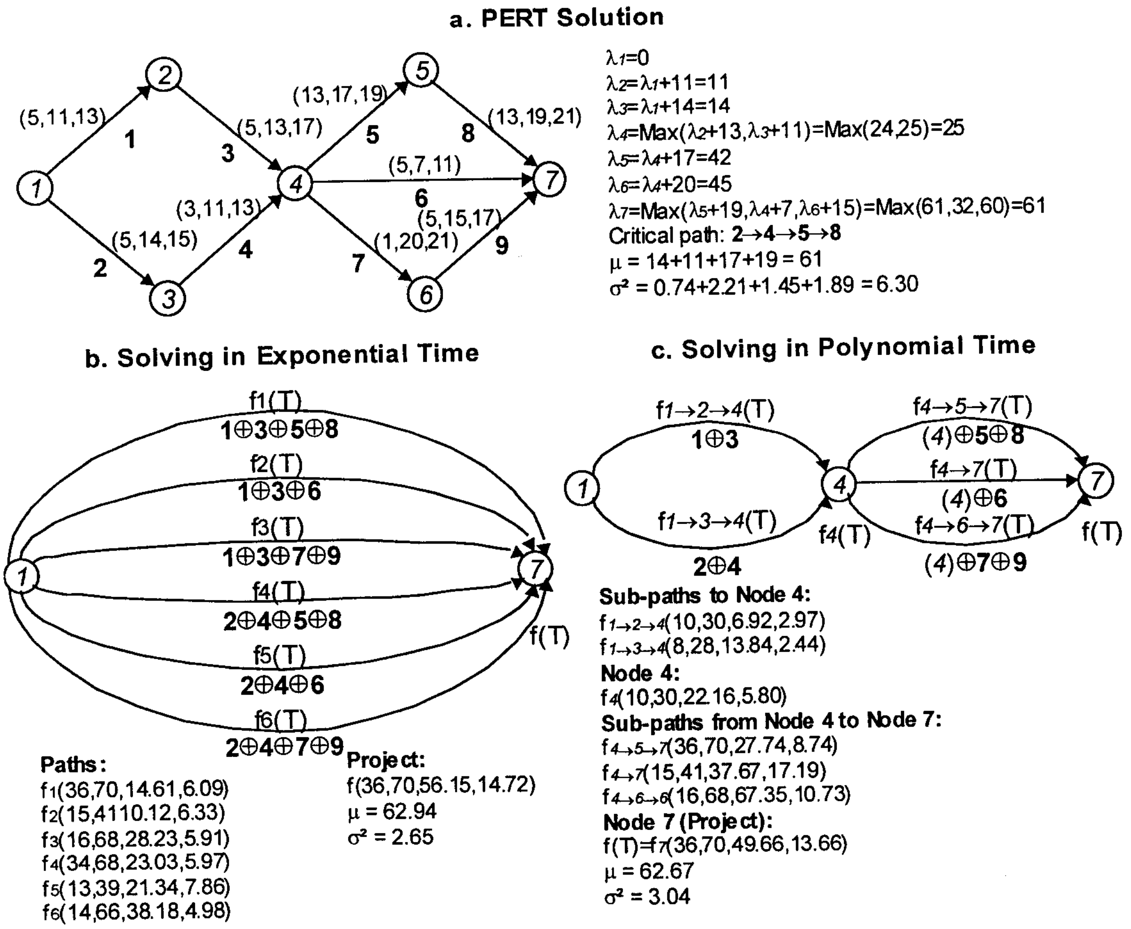


Figure 8. 7-Nodes/9-Activities Example.

The same system of equations can be applied to solve the problem in a polynomial maximum number of steps (5 in total for this example) by considering all nodes that have more than one incoming arc. Node 4 is the first such node. There are two paths before node 4. The first path follows nodes *1*, *2*, and *4* corresponding to activities **1** and **3**. The beta distribution describing the duration of path *1→2→4* (also path **1→3**) is $f_{1\rightarrow 2\rightarrow 4}(T)$ with parameters 10, 30, 6.92, and 2.97, given by the stochastic sum of activities **1⊕3** according to equations (25) to (31). The same applies to path *1→3→4*. The completion time at node 4, obtained by the joint integration of paths *1→2→4* and *1→3→4* according to equations (25) to (31) is $f_4(T)$. There are 3 alternative paths between nodes 4 and 7. The first path is *4→5→7* with a duration time given by the completion time at node 4 plus the duration of activities 5 and 8, denoted as (*4*)⊕5⊕8. The second and third paths are *4→7* and *4→6→7*. The joint numerical integration of these three paths yields the parameters for the beta distribution describing the completion time of the project, which has mean and variance of 62.67 and 3.04, respectively. These differ slightly from the brute force results due to rounding error. The range parameters and the shape parameters of the corresponding beta distribution are a=36, b=70, α=49.66, and β=13.66. The total number of sub-paths in this case of polynomial complexity is 2+3=5.

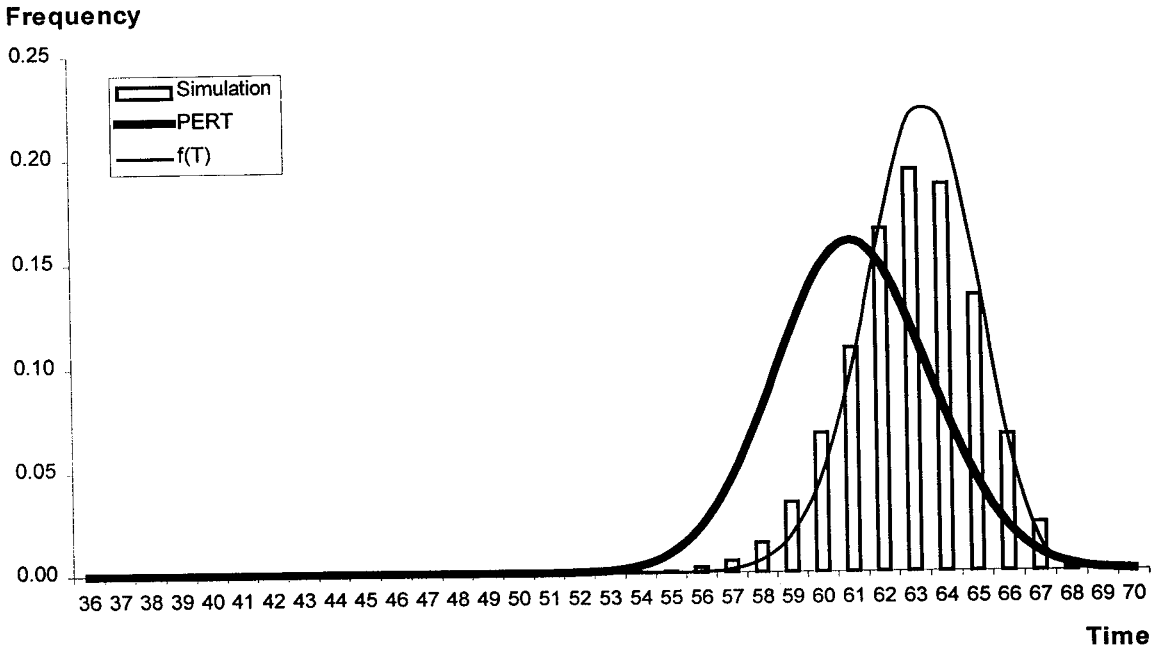


Figure 9. PERT versus Stochastic Sum/Integration.

The latter is the theoretical mean and variance of the project completion time, assuming independent beta distributions for activity duration times. The corresponding probability density function, assuming a beta distribution, is shown in Figure 9 as f(T). To test the accuracy of these results through simulation, a total of 25,000 activity duration times were generated at random, yielding an average of 62.42 and a variance of 4.22. Compare the PERT estimate (61) versus the average project completion times obtained by stochastic sum/joint integration (62.94 for the exponential complexity problem and 62.67 for the polynomial complexity problem). Clearly, the latter are closer to the theoretical approximation obtained by extensive Monte Carlo simulation (62.42). Figure 9 compares the results from the three approaches: a histogram of the results obtained from the simulation, the probability density functions describing project completion times obtained according to PERT, as well as the results obtained according to our methodology.

2.11 Discussion

The simulation results plotted in the histogram of Figure 9 constitute a statistical sample of the theoretical distribution describing the project completion time. Unfortunately, even extensive simulation yields only approximate results. In each run, a set of activity duration times following beta density functions is generated at random. The project completion time for that run is calculated according to the PERT/CPM forward pass calculation portrayed in equation (19) by assuming such duration times are deterministic, so that the actual duration time as opposed to the mean is used. But in fact, activity duration times are randomly distributed variables described by a continuous beta density function. Consequently, only the histogram of a simulation with an infinite number of runs would yield the theoretical distribution describing the project completion time. Nevertheless, a simulation with 25,000 sets of activity duration times generated at random can be considered close enough to the theoretical distribution. As we can see in Figure 9, the normal distribution obtained according to PERT is a poor approximation to the simulation results. But the beta density function obtained according to our procedure is a much better match.

In this particular example, path duration times are in fact independent if we integrate the completion time at node 7 by considering a start time at node 4 of zero and then adding the result obtained at node 7 to the result obtained at node 4. The mean completion time obtained in that case is also 62.67 with a variance of 3.11, slightly larger than the variance of 3.04 shown at the bottom of Figure 8c. The shape parameters also change slightly, from 49.66 to 48.51 for α and from 13.66 to 13.34 for β. The probability density function of the latter looks almost exactly the same as the one shown as f(T) in Figure 9. Undoubtedly, the distribution of the population need not follow a beta density function, but if that is assumed to be the case, f(T) in fact fits well the project completion time. For practical purposes, f(T) can be considered a good approximation to the distribution of the theoretical project completion time when sub-paths do not share one or more activities.

The recommendation to the practitioner is to avoid calculating the mean and the variance according to the PERT textbook formula [89], $\mu=(a+4m+b)/6$ where m is the mode and $\sigma^2=((b-a)/6)^2$, and then simply calculating the maximum mean and variance at each node. In fact, the textbook formula assumes a fixed value for the sum of the shape parameters ($\alpha+\beta=4$) to calculate the mean, and it calculates the variance as an approximation to that assumption. Furthermore, PERT does not consider the variance when determining which path is the longest, since the variance of the project completion time is assumed to be the same as the variance of the path with the longest sum of mean duration times. All these assumptions typically lead to optimistic planning due to less than actual project completion times. Instead, the practitioner could use the beta density function for the completion time of the sub-path with the maximum mean at each node and simply apply equations (25), (26), (11), and (12) to directly obtain the range and shape parameters. This avoids calculating the theoretical mean and variance at each node by integrating as indicated in equations (30) and (31), while at the same time providing much better results for considering the shape and variance of each activity, and the shape and variance as a function of the absolute bounds for each node.

Chapter 3. Project Tradeoffs

3.1 Introduction

Although the relationships between project management dimensions vary from time to time and from project to project, a systemic approach can be used to elucidate the nature of the underlying tradeoffs [44][45][79].

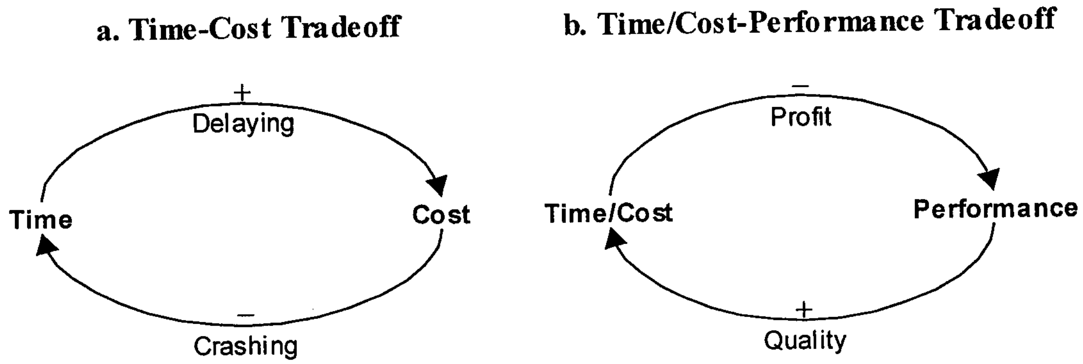


Figure 10. Time, Cost, and Performance Tradeoffs.

Figure 10a pictures the systemic relationship between time and cost using influence diagrams. If the project is delayed it costs more so that there is a positive correlation between time and cost. But in order to deliver on time, additional resources can be directed to critical activities while keeping resources to a minimum for non-critical activities. The latter, called crashing, constitutes a negative correlation between cost and time [89]. The existence of both a positive and a negative correlation between time and cost imply the existence of an equilibrium in which an optimal project completion time is achieved at a minimum cost. Figure 10b pictures how the time/cost tradeoff is further influenced by performance. Improving the quality of the product requires investing more resources, which increases cost and will increase time if such resources are limited. But if more resources are invested and it takes longer to complete the project or it costs more, the Internal Rate of Return (IRR) of the project measuring its profitability is reduced. Consequently, there must be an optimal time/cost tradeoff that yields optimal project performance as measured by its IRR.

3.2 Time-Cost Tradeoff with Unlimited Resources

Costs are a function of both project completion time and criticality. For budgeting purposes, there are two types of costs: direct costs for each activity and indirect costs for the overall project. Direct costs include the cost of the material, equipment, and direct labor required to perform an activity. When the activity is subcontracted, the direct cost is equal to the price of the subcontract. Indirect costs include in addition to supervision and other overhead costs, the interest charges on the cumulative project investment and overdue penalty costs [5][17][58][67][80].

Let v_k (or v_{ij}) denote the all-crashed direct cost of activity k in arc (i,j), which occurs when the activity duration time is minimum ($t_k=a_k$ or equivalently, $t_{ij}=a_{ij}$) and u_k (or u_{ij}) be the all-delayed direct cost for the maximum duration time ($t_k=b_k$ or equivalently, $t_{ij}=b_{ij}$). All-crashed is the minimum activity duration time achieved by crashing all the way the duration of the activity at the highest total cost. All-delayed is the maximum activity duration time when resources are used at a minimum pace, which results in the lowest total cost. Also, let d_k (or d_{ij}) be the direct cost of activity k in arc (i,j). For simplicity, crashing, as pictured in Figure 10a, can be assumed to be an inversely proportional relationship between time and cost [6][35], as portrayed by equation (43).

$$d_k \propto \frac{1}{t_k} \tag{43}$$

Reducing duration requires increasing the direct cost of the activity. The proportionality in equation (43) can be transformed into equality by adding a proportionality factor and a constant. Let χ_k and ϕ_k be such constants as shown in equation (44).

$$d_k = \frac{\chi_k}{t_k} + \phi_k \tag{44}$$

Since the direct cost should equal v_k when t_k=a_k and it should equal u_k when t_k=b_k, equation (42) is transformed into the system of equations (44a) and (44b).

$$v_k = \frac{\chi_k}{a_k} + \phi_k \tag{44a}$$

$$u_k = \frac{\chi_k}{b_k} + \phi_k \tag{44b}$$

Solving yields χ_k=$a_k b_k(v_k-u_k)/(b_k-a_k)$ and ϕ_k=$(b_k u_k-a_k v_k)/(b_k-a_k)$. Substituting χ_k and ϕ_k into equation (44) yields equation (45).

$$d_k = \frac{a_k b_k}{t_k}\left(\frac{v_k - u_k}{b_k - a_k}\right) + \left(\frac{b_k u_k - a_k v_k}{b_k - a_k}\right) \tag{45}$$

Let V and U denote the maximum and minimum direct costs of the project defined as the sum of the maximum and minimum direct costs of each and every activity in the network according to equations (46) and (47).

$$U = \sum_{k=1}^{m} u_k \tag{46}$$

$$V = \sum_{k=1}^{m} v_k \tag{47}$$

Also, let T_n be the project's completion time obtained according to equation (1) and let a, μ, and b denote the minimum, average, and maximum project completion times, respectively. Just as in equation (45), the total direct cost of the project, D, is inversely proportional to the project completion time, T, as shown in equation (48).

$$D = \frac{ab}{T}\left(\frac{V-U}{b-a}\right) + \left(\frac{bU-aV}{b-a}\right) \tag{48}$$

The project indirect cost, I, is directly proportional to the project completion time, as indicated by equation (49).

$$I \propto T \tag{49}$$

The proportionality in equation (49) can be transformed into equality by adding proportionality constants. Let O denote the minimum overhead (indirect) cost at time T=0, and ΔO denote the increment (slope) of the total indirect cost as a function of time. Then, equation (50) indicates the directly correlated relationship between time and cost as pictured in the influence diagram of Figure 10a.

$$I = O + \Delta OT \tag{50}$$

Figure 11a plots direct and indirect project costs. The total cost, C, equals the sum of both direct and indirect costs as shown in equation (51).

$$C = I + D = O + \Delta OT + \frac{ab}{T}\left(\frac{V-U}{b-a}\right) + \left(\frac{bU-aV}{b-a}\right) \tag{51}$$

Let C_a and C_b denote the costs corresponding to the minimum and maximum project completion times (T=a and T=b), respectively. Then, from equation (51) we have that C_a and C_b

are given according to equations (51a) and (51b), which is exactly what we expect given that V and U are the all-crashed (for T=a) and all-delayed (for T=b) project direct costs, respectively.

$$C_a = O + \Delta Oa + V \tag{51a}$$

$$C_b = O + \Delta Ob + U \tag{51b}$$

The maximum cost, C_{Max}, is either C_a or C_b since these correspond to the extreme values of Figure 11b, as shown in equation (51c).

$$C_{Max} = \text{Max}\{C_a, C_b\} \tag{51c}$$

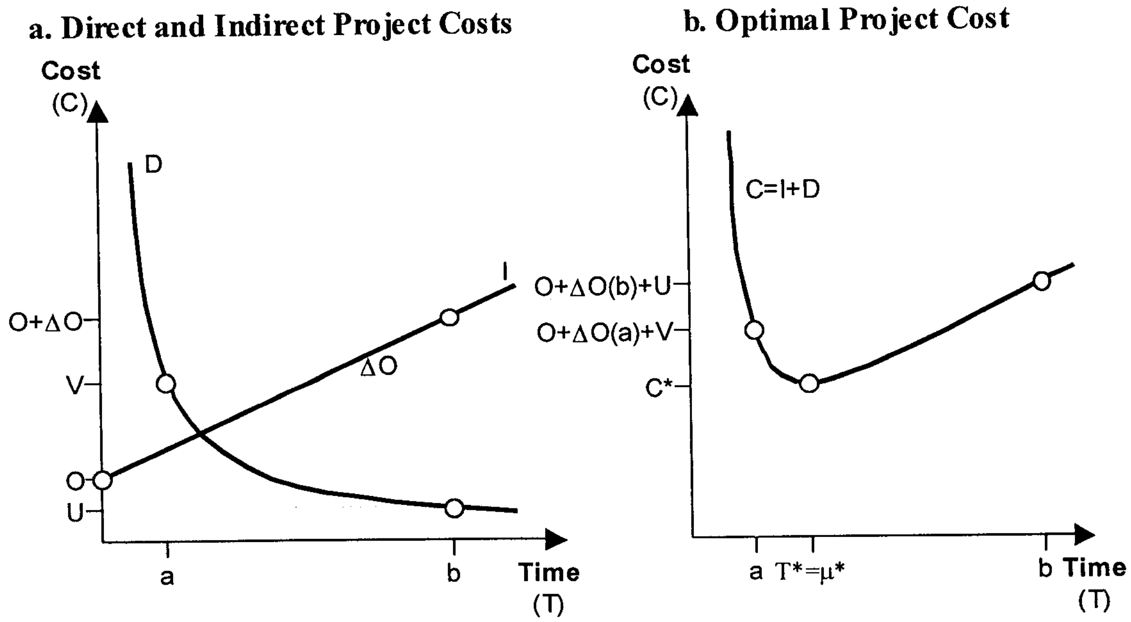


Figure 11. Time-Cost Tradeoffs.

Clearly, the minimum cost (C_{Min}=C*) occurs when the slope (derivative of the cost with respect to time) of the total cost curve is zero as shown in equation (52).

$$\frac{\partial C}{dT} = \Delta O - \frac{ab}{T^2}\left(\frac{V-U}{b-a}\right) = 0 \tag{52}$$

But since completion time is a random variable with known probability density function, cost must also be a random variable with a probability density function described as a function of time, where $C_a<C<C_b$. This means that the optimal time-cost tradeoff obtained when equation (52) equals zero is in fact the optimal of the average (expected) project completion time. Let T* denote such optimal project completion time and $C_{Min}=C^*$ the corresponding optimal (and minimum) project cost. Then, T* is given according to equation (53) and $C_{Min}=C^*$ is given by equation (51), calculated by replacing T for T*.

$$T^* = \sqrt{ab\left(\frac{V-U}{\Delta O(b-a)}\right)} \tag{53}$$

Although equation (53) applies to the overall project, it can be used to compute the optimal time-cost tradeoff of each activity by calculating the minimum and maximum completion time as well as the minimum and maximum direct and overhead costs of each activity.

3.3 Calculating the Optimal Schedule

Cost is additive, so that direct costs can be summed throughout the activity network. Let V_j and U_j denote the maximum and minimum cumulative direct costs of node j occurring at minimum and maximum completion times A_j and B_j, respectively. Also, let V_{ij} and U_{ij} be the maximum and minimum cumulative direct costs of activity k in arc (i,j). By definition, the maximum and minimum cumulative direct cost of the first (starting) node are zero as shown in equations (54) and (55).

$$U_1 = 0 \tag{54}$$

$$V_1 = 0 \tag{55}$$

The minimum and maximum cumulative direct costs are given as the sum of the costs of minimum and maximum costs of all preceding activities as shown in equations (56) and (57).

$$U_j = u_{12} + \ldots + u_{ij}, j{=}2,\ldots,n \tag{56}$$

$$V_j = v_{12} + \ldots + v_{ij}, j{=}2,\ldots,n \tag{57}$$

Also, the minimum and maximum cumulative costs of node j are given according to equations (58) and (59).

$$U_j = U_i + \sum_{i<j} u_{ij}, \ i{=}1,\ldots,n\text{-}1, j{=}2,\ldots,n \tag{58}$$

$$V_j = V_i + \sum_{i<j} v_{ij}, \ i{=}1,\ldots,n\text{-}1, j{=}2,\ldots,n \tag{59}$$

The minimum and maximum completion times, A_j and B_j, are given according to equations (25) and (26). Also, let D_j and T_j denote the cumulative direct cost and completion time at node j, respectively, so that $D_j=V_j$ when $T_j=A_j$ and $D_j=U_j$ when $T_j=B_j$. If I_j denotes the indirect cost assigned to node j and C_j denotes the total cost of node j, then equation (51) can be rewritten as equation (60) to portray the total cumulative cost at node j.

$$C_j = I_j + D_j = O + \Delta O(T_j) + \frac{A_j B_j}{T_j}\left(\frac{V_j - U_j}{B_j - A_j}\right) + \left(\frac{B_j U_j - A_j V_j}{B_j - A_j}\right) \tag{60}$$

Taking the derivative of C_j from equation (60) with respect to T_j and equating to zero yields the optimal average completion time of node j, T_j*, as shown in equation (61). (By definition, the optimal average completion time of the starting node is zero, T_1*=0.)

$$T_j^* = \sqrt{A_j B_j\left(\frac{V_j - U_j}{\Delta O(B_j - A_j)}\right)}, j{=}2,\ldots,n \tag{61}$$

Let A_{ij} and B_{ij} be the minimum and maximum completion times of activity k in arc (i,j) given according to equations (62) and (63), where a_{ij} and b_{ij} are the corresponding minimum and maximum duration times. Notice that $A_{ij} = A_i + a_{ij}$, $B_{ij} = B_j \geq B_i + b_{ij}$, because although activity (i,j) cannot start before T_i, it need not start at time T_i and may be delayed. Nevertheless, the maximum delay allowed cannot exceed the maximum completion time at node j. Consequently, the minimum completion time of the activity is the minimum completion time of node i plus the minimum duration time of activity (i,j) as shown in equation (62).

$$A_{ij} = A_i + a_{ij} \tag{62}$$

However, its maximum completion time may be greater than the maximum completion time of node i plus the maximum duration time of activity (i,j) but cannot be greater than the maximum completion time of node j, so that the maximum completion time of activity (i,j) is in fact defined as the maximum completion time of node j as shown in equation (63).

$$B_{ij} = B_j \tag{63}$$

Also, let λ_{ij} be the expected (average) completion time of activity k in arc (i,j), given according to equation (64), where μ_i is the expected (average) completion time at node i.

$$\lambda_{ij} = \mu_i + \mu_{ij} \tag{64}$$

Then, the optimal expected completion time of activity (i,j), $\lambda_{ij}*$, is given according to equation (65).

$$T_{ij}^* = \lambda_{ij}^* = \sqrt{A_{ij}B_{ij}\left(\frac{V_{ij} - U_{ij}}{\Delta O(B_{ij} - A_{ij})}\right)}, \text{ i=1,...,n-1, j=2,...,n} \tag{65}$$

If activity (i,j) is not one hundred percent critical, there is a slack (time interval) between its completion time (λ_{ij}) and the completion time of node j (μ_j). Let S_{ij} denote such slack. Clearly, the slack is the difference between the completion time at node j and the completion time of activity (i,j) as shown in equation (66).

$$S_{ij} = \mu_j - \lambda_{ij} \tag{66}$$

But activity (i,j) may be delayed if it is not on the critical path, that is, it needs not be scheduled immediately after the average completion time at node i, which indicates the average completion time of all preceding activities. The maximum average delay is given by the activity's slack, so that the actual average delay must be a percentage of that. Let $0<\gamma_{ij}<1$ denote the percentage of the slack by which activity (i,j) has been delayed, so that the average time at which activity (i,j) starts is $\gamma_{ij}S_{ij}$ time units after the completion of node i.

Solving for μ_j from equation (66) and substituting λ_{ij} from equation (64) yields equation (67).

$$\mu_j = \mu_i + \mu_{ij} + S_{ij} \tag{67}$$

Also, since γ_{ij} is a fraction of S_{ij}, we have that S_{ij} can be divided in two parts, $\gamma_{ij}S_{ij}$ and $(1-\gamma_{ij})S_{ij}$, as shown in equation (68).

$$S_{ij} = \gamma_{ij}S_{ij} + (1-\gamma_{ij})S_{ij} \tag{68}$$

Substituting S_{ij} from equation (68) into equation (67) yields equation (69).

$$\mu_j = \mu_i + \gamma_{ij}S_{ij} + \mu_{ij} + (1-\gamma_{ij})S_{ij} \tag{69}$$

Figure 12a illustrates equation (67) along the horizontal time coordinate (T). In Figure 12a, activity (i,j) is scheduled as soon as possible so that its expected completion time (λ_{ij}) equals the expected completion time of node i (μ_i) plus the activity's expected duration (μ_{ij}), so that μ_{ij} is the distance between λ_{ij} and μ_i. Since the expected completion time of the activity cannot be greater than the expected completion time of node j, the activity's slack is given by the distance between μ_j and λ_{ij}.

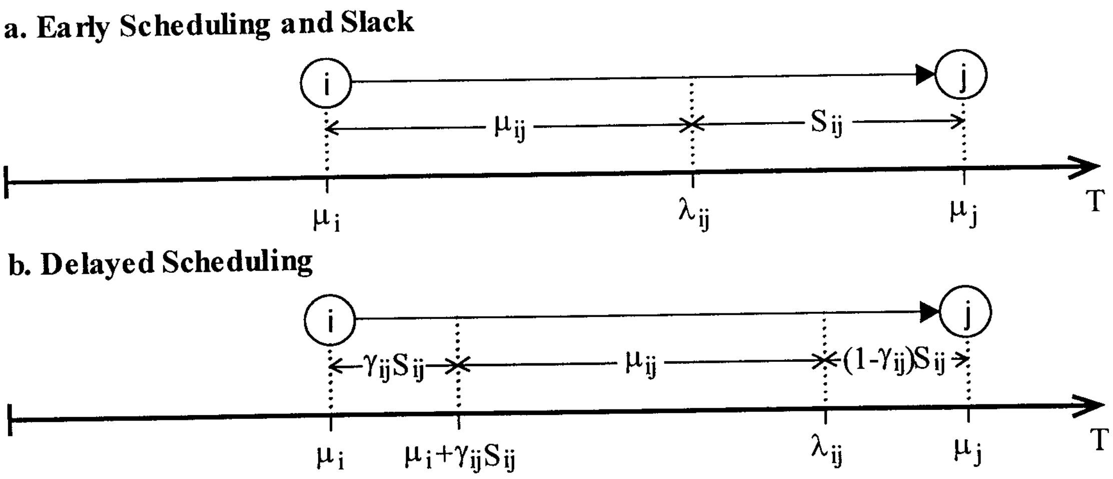


Figure 12. Completion Time Slack and Delay.

Figure 12b illustrates equation (69). In Figure 12b the expected completion times of nodes i and j and the expected duration time of activity (i,j) are the same, but the expected time at which the activity starts is not the completion time of node i but the latter plus the activity's delay given by a fraction of its slack, γ_{ij}. Thus, the expected start time of activity (i,j) is the expected completion time of node i (μ_i) plus the activity's delay ($\gamma_{ij}S_{ij}$). Also, the expected completion time of the activity (λ_{ij}) is given by its expected start time ($\mu_i+\gamma_{ij}S_{ij}$) plus its expected duration time (μ_{ij}). Since the distance between the completion times of nodes i and j ($\mu_j-\mu_i$) is the same as in Figure 12a, the distance between λ_{ij} and μ_j must be $(1-\gamma_{ij})S_{ij}$ so that the total slack does not change ($\gamma_{ij}S_{ij}+(1-\gamma_{ij})S_{ij}=S_{ij}$) as shown in equation (68). The system of equations (70) and (71) portrays Figure 12b.

$$\lambda_{ij} = \mu_i + \gamma_{ij}S_{ij} + \mu_{ij} \tag{70}$$

$$\lambda_{ij} = \mu_j - (1-\gamma_{ij})S_{ij} \tag{71}$$

The optimal of the expected completion time of node j (and i) is calculated according to equation (61), whereas the optimal of the expected completion time of activity (i,j) is given by equation (65). Thus the system of equations (70) and (71) can be expressed in terms of these optimal solutions by substituting λ_{ij} for λ_{ij}*, μ_i for μ_i*, and μ_j for μ_j*, where μ_{ij}* and γ_{ij}* are unknowns to be found as shown in equations (72) and (73).

$$\lambda_{ij}^* = \mu_i^* + \gamma_{ij}^* S_{ij} + \mu_{ij}^* \tag{72}$$

$$\lambda_{ij}^* = \mu_j^* - (1 - \gamma_{ij}^*) S_{ij} \tag{73}$$

Solving for γ_{ij}* from equation (73) yields equation (74).

$$\gamma_{ij}^* = 1 - \frac{\mu_j^* - \lambda_{ij}^*}{S_{ij}} \tag{74}$$

Then, substituting γ_{ij}* from equation (74) into equation (72) yields equation (75).

$$\mu_{ij}^* = \mu_j^* - \mu_i^* - S_{ij} \tag{75}$$

With this optimal expected duration time of each activity, a new project completion time can be obtained according to the system of equations discussed in chapter 2, and a budget for the overall project calculated according to equations (46) to (53).

3.4 Performance Tradeoffs

Although costs are additive (i.e., the total cost is the sum of the cost of each activity), they do not occur at the same time. Depending on the accounting method used, costs could be allocated at the beginning (First-In, First-Out or FIFO), at the end (Last-In, First-Out or LIFO), or even at the middle point between the beginning and the end of the activity to which such costs are assigned. For convenience, let us assume the costs are allocated as soon as the activity starts (FIFO). In FIFO, indirect costs are allocated at the beginning of the project. The project starts at time $T_1=0$, so that the project's indirect cost (I) given according to equation (50) occurs at time $T_1=0$. Let $d_{ij}=d_k$ be the direct cost given according to equation (76), obtained by substituting k for ij from equation (45), where $v_{ij}=v_k$ is the all-crashed direct cost, $u_{ij}=u_k$ is the all-delayed direct cost, and $\mu_{ij}*$ is the optimal value of the expected duration time of activity k in arc (i,j) calculated according to equation (75) with a minimum and a maximum duration time of a_{ij} and b_{ij}.

$$d_{ij} = \frac{a_{ij}b_{ij}}{\mu^*_{ij}}\left(\frac{v_{ij}-u_{ij}}{b_{ij}-a_{ij}}\right)+\left(\frac{b_{ij}u_{ij}-a_{ij}v_{ij}}{b_{ij}-a_{ij}}\right) \tag{76}$$

Also, let C_i be the direct cost allocated to node i occurring on average at time $T_i=\mu_i*$ for all i=1,...,n-1. Then, C_i is the sum of the direct costs of all the activities coming out of node i as indicated in equation (77).

$$C_i = \begin{cases} I+\sum\limits_{i<j} d_{ij}, & i=1 \\ \sum\limits_{i<j} d_{ij}, & i=2,\ldots,n-1 \end{cases} \tag{77}$$

Assuming the planning horizon for the project is given by H and that Q denotes the project's fractional Internal Rate of Return (IRR), the project's Net Present Value (NPV) is given according to equation (78), where E is the income per time unit expected after the completion time [14].

$$\text{NPV} = -\sum_{i=1}^{n-1}\frac{C_i}{(1+Q)^{T_i}}+\left(\frac{E}{(1+Q)^{T_n}}\right)\left(\frac{1-1/(1+Q)^{H-T_n}}{Q}\right) \tag{78}$$

Figure 13 illustrates the arrangement of negative cash flows (costs, between $T=T_1=0$ and $T=T_n$) and positive cash flows (income, between $T=T_n$ and $T=H$) through time. Figure 13a shows the activity network corresponding to the cash flows diagram of Figure 13b. Notice that each cost C_i occurring at time T_i indicates the completion of node i and the beginning of each and every activity coming out of node i.

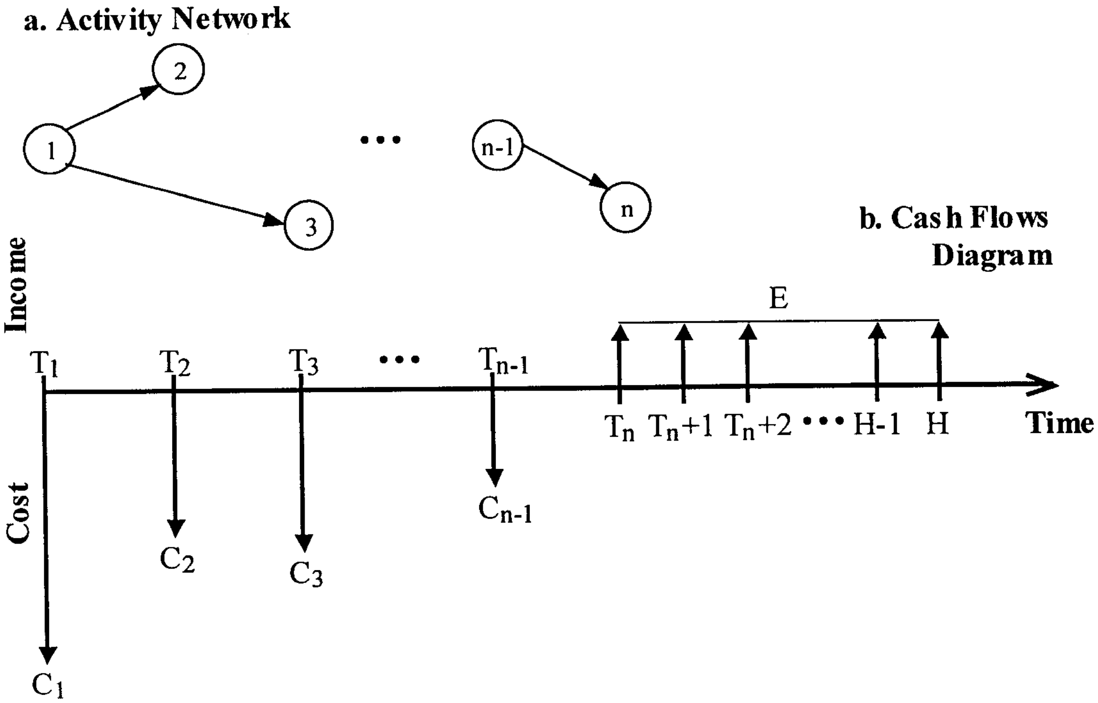


Figure 13. Cost and Income through Time.

The first term in equation (78) is the sum of each cost C_i occurring at time T_i discounted at the Internal Rate of Return (IRR) of the project, whereas the second term is the formula for an annuity of H-T_n periods occurring at time T_n also brought to the present.

Figure 13 implicitly assumes that time is expressed in non-negative integer units such as days so that annuity inflow occurs at subsequent time periods (T_n, T_{n+1}, ..., H-1, H). Equation (79) expresses the annuities as a sequence of H-T_n cash inflows and is equivalent to equation (78).

$$\mathrm{NPV} = -\sum_{i=1}^{n-1} \frac{C_i}{(1+Q)^{T_i}} + \frac{E}{(1+Q)^{T_n}} + \frac{E}{(1+Q)^{T_n+1}} + \ldots + \frac{E}{(1+Q)^{H-1}} + \frac{E}{(1+Q)^{H}} \qquad (79)$$

The Net Future Value (NFV) is the projection of the project's value into the future, that is, into the time at which the last cash flow occurs (T=H), as shown in equation (80).

$$\mathrm{NFV} = -\sum_{i=1}^{n-1} C_i (1+Q)^{H-T_i} + E(1+Q)^{H-T_n} + E(1+Q)^{H-T_n-1} + \ldots + E(1+Q) + E \qquad (80)$$

The relationship between NPV and NFV is given according to equations (81a) and (81b).

$$\mathrm{NPV} = \frac{\mathrm{NFV}}{(1+Q)^{H}} \qquad (81a)$$

$$\mathrm{NFV} = \mathrm{NPV}(1+Q)^{H} \qquad (81b)$$

By definition [14], the Internal Rate of Return (IRR) of the project is the discount rate Q at which the NPV is zero, and since from equation (81b) NFV is $(1+Q)^H$ times NPV, NFV is also zero when NPV equals zero. Therefore, the IRR is the value Q for which equation (80) equals zero as shown in equation (82).

$$-\sum_{i=1}^{n-1} C_i (1+Q)^{H-T_i} + E(1+Q)^{H-T_n} + E(1+Q)^{H-T_n+1} + \ldots + E(1+Q) + E = 0 \qquad (82)$$

Equation (82) is a polynomial in (1+Q) of degree H. Substituting (1+Q) for X from equation (82) yields equation (83).

$$-C_1X^{H} - \ldots - C_{n-1}X^{H-T_{n-1}} + EX^{H-T_n} + EX^{H-T_n+1} + \ldots + EX + E = 0 \qquad (83)$$

Applying the Newton-Raphson numerical method [12] yields the root (value of X=1+Q) for which equation (82) equals zero, so that the project's IRR is given according to equation (84). Although multiple roots are in principle possible, equation (83) would not lead to multiple roots because there is only one change in sign of the series of cash flows [13].

$$\mathrm{IRR} = \mathrm{Q} = \mathrm{X} - 1 \tag{84}$$

3.5 Example

Consider the example from chapter 2. The overhead cost is $2,500 and the overhead cost per time unit is $100 per week. Table 1 contains all additional data. The last column of Table 1 indicates the optimal time-cost tradeoff point according to equation (53). Notice that in some cases, the optimal tradeoff point is outside the boundaries (a and b). When that happens (see activities 3, 5, 8, and 9), the boundary is used as the mean duration time.

Activity	**a**	**b**	**U**	**V**	μ^*
1	5	13	$250	$700	6.05
2	5	15	$200	$750	6.42
3	5	17	$200	$500	~~4.61~~ = 5
4	3	13	$350	$800	4.19
5	13	19	$1,000	$2,500	~~24.85~~ = 19
6	5	11	$300	$900	7.42
7	1	21	$400	$600	1.45
8	13	21	$150	$450	~~10.12~~ = 13
9	5	17	$650	$850	~~3.76~~ = 5

Table 1. Time-Cost Tradeoff Example.

The expected weekly income is $3,000 after the project is complete, and the planning horizon is 60 weeks. Costs occur at each node. The list of cash flows due to costs or income is given in Table 2. The Internal Rate of Return (our measure of performance) that corresponds to the latter is 3.61% and the Net Present Value (NPV) for a Minimum Attractive Rate of Return (MARR) of 3% is $2,223.37.

Time	Cash Flow	Time	Cash Flow	Time	Cash Flow
0	$-2,500	45	$3,000	53	$3,000
6.05	$-1,178.09	46	$3,000	54	$3,000
6.42	$-1,209.52	47	$3,000	55	$3,000
11.04	$-2,052.85	48	$3,000	56	$3,000
30.05	$-2,900	49	$3,000	57	$3,000
12.50	$-679.83	50	$3,000	58	$3,000
43.05	$-4,383.24	51	$3,000	59	$3,000
44	$3,000	52	$3,000	60	$3,000

Table 2. Cash Flow Calculations for Example 1.

Now consider a larger software development project example[2] consisting of 15 activities. The overhead cost is $10,000 and the overhead cost per day is $100. The planning horizon is 80 days and the income per day after project completion is $5,000. Note that the costs assumed for this example are our estimates. Table 3 summarizes all the data. The sequence of node completion times and costs is given in Table 4. The project's IRR is 3.63% and the project's NPV for a MARR of 3% is $6,055.16.

In practice, it is unlikely that either project 1 or 2 would be undertaken due to their low internal rates of return. Another factor that enters into choosing a project is its risk. This is discussed in the next chapter, in conjunction with rate of return and project classifications such as mandatory or mutually exclusive.

[2] http://www.thebusinessmac.com/features/projmgmt5.shtml

Activity		Predecessors	a	b	U	V	μ^*
1	Needs Analysis		9	12	100	250	~~7.35~~ = 9
2	Specifications	1	5	8	300	750	7.75
3	Select Server	2	4	13	125	275	~~2.94~~ = 4
4	Select Software	2	7	16	50	400	~~6.60~~ = 7
5	Select Cables	3	3	7	30	90	~~1.77~~ = 3
6	Purchasing	4,5	2	6	75	225	2.12
7	Manuals	4,5	6	11	155	235	~~3.25~~ = 6
8	Wire Offices	6	7	15	100	350	~~5.73~~ = 7
9	Set Up Server	6	4	6	95	175	~~3.10~~ = 4
10	Develop Training	7	12	18	25	165	~~7.10~~ = 12
11	Install Software	9	3	7	45	200	~~2.85~~ = 3
12	Connect Network	8,11	2	9	35	155	~~1.76~~ = 2
13	Train Users	10,12	7	12	185	550	7.83
14	Test/Debug	12	11	18	355	700	~~9.88~~ = 11
15	Acceptance	13,14	2	8	245	445	2.31

Table 3. Example 2 Data.

Time	Cash Flow	Time	Cash Flow	Time	Cash Flow
0.00	-10,000	53.00	5,000	67.00	5,000
9.00	-1,119.69	54.00	5,000	68.00	5,000
16.75	-1,099.19	55.00	5,000	69.00	5,000
20.75	-647.117	56.00	5,000	70.00	5,000
23.75	-1,437.68	57.00	5,000	71.00	5,000
25.87	-424.264	58.00	5,000	72.00	5,000
29.87	-554.677	59.00	5,000	73.00	5,000
32.87	-1,526.17	60.00	5,000	74.00	5,000
29.75	-708.923	61.00	5,000	75.00	5,000
34.87	-352.038	62.00	5,000	76.00	5,000
41.75	-1,164.86	63.00	5,000	77.00	5,000
49.58	-3,028.71	64.00	5,000	78.00	5,000
51.89	-640.214	65.00	5,000	79.00	5,000
52.00	5,000	66.00	5,000	80.00	5,000

Table 4. Cash Flow Calculations for Example 2.

3.6 Discussion

It is important to notice that the optimal time-cost tradeoff could have been calculated using cumulative costs as opposed to the cost of each activity. In such a case, care should be taken to avoid the addition of the same cost several times.

As we have seen in the previous examples, the completion time at each node has a direct impact on performance since that indicates the time at which the cost allocated to any given node occurs. Table 2 summarizes the project. But there are many different cash flow tables that can be created for our example. We could generate the cash flows for all minimum completion times, or the table for all maximum completion times. The existence of this variability is due to the uncertain nature of our time estimates. Notice that there were more cases of optimal completion times (μ*) outside bounds in the second example than in the first example. This is due to tighter bounds in example 2.

The minimum and maximum project performance is measured by the project's minimum and maximum IRR (Q_{Min} and Q_{Max}). The minimum IRR is the rate of return obtained by solving equation (84) for the maximum completion time and maximum cost at each node. The maximum IRR typically is the one corresponding to the optimal completion time and cost at each node. Nevertheless, that need not necessarily be the case because the minimum (all-crashed) completion time may provide a higher rate of return despite the higher cost, by allowing the positive cash flows to occur sooner. Therefore, it is important to evaluate both possibilities in order to obtain the range in which the IRR is expected to vary.

Having minimum and maximum estimates for costs and IRR allow us to estimate cost and IRR as an expected value (average of the maximum and minimum estimates) with the corresponding standard deviation if we assume cost and IRR to follow a beta distribution.

The next step after being able to find the time, cost, and performance estimates of individual projects is to use these results in considering groups of projects, also called portfolios, and how to decide which projects to include into the portfolio.

Chapter 4. Portfolio Selection

4.1 Introduction

Project selection is one of the first and most critical activities in project management. Deciding from a pool of available and competing projects which ones should be undertaken (thus assigning limited resources to them) and which ones should not be undertaken or terminated is a complex decision. Overall value maximization, balance among dimensions, and business strategy should be considered. The very essence of portfolio management portrayed by Cooper *et al.* [20] as a "dynamic decision process… constantly up-dated and revised… [where] new projects are evaluated, selected and prioritized; existing projects may be accelerated, killed or de-prioritized; and resources are allocated or re-allocated to the active projects" increases the difficulty. Furthermore, portfolio selection is a process characterized by uncertainty and changing information: new opportunities arise, multiple goals as well as strategic considerations are required, interdependence among projects (either when competing for scarce resources or when synergies are achieved) exist, not to mention multiple decision-makers and locations. Consequently, a mathematical model built into a flexible Group Decision Support System (GDSS) developed within an optimally designed Web-based User Interface (WUI) to foster interaction between decision-makers seems to be the best long term approach to tackle such a complex decision making process. Appendix C discusses the design of the interface for the portfolio selection problem.

According to Meredith and Mantel [58] project selection methods can be classified as nonnumeric (qualitative) or numeric (quantitative). The sacred cow, operating necessity, competitive necessity, product line extension, and the comparative benefit model are among the qualitative methods. Profitability models (payback period, average rate of return, NPV, IRR, profitability index, as well as others that subdivide the elements of the cash flow, include terms of risk or uncertainty or consider the effect on other projects or the organization) and scoring models (weighted and non-weighted zero-one factor models with or without constraints usually solved using integer programming as well as goal programming when multiple objectives are given) are among the quantitative methods.

A decision support system for project portfolio selection is presented by Archer and Ghasenzadeh [3]. Our portfolio selection model is a maximization zero-one integer programming scoring model that is more extensive because it explicitly considers risk. For an alternative zero-one integer programming model, refer to work by Ghasemzadeh, Archer and Iyogun [36].

4.2 Zero-one Integer Programming Model for Portfolio Selection

There is no such thing as the optimal portfolio when we consider the tradeoffs among time, cost, and performance (not to mention risk preferences). Decision-makers have to weigh multiple project dimensions and intuitively decide how adding or removing a specific project would have an impact on the portfolio. In other words, they face intuitive decisions on marginal contribution (gain or loss). Our conjecture is that *the best decision is achieved when overall cost and time are minimized while maximizing performance for a given risk profile.* The question is how to translate this qualitative statement into a quantitative model that can be optimized. It is important to realize that the solution given by the model depends on the accuracy of the time, cost and performance estimates for individual projects that are provided by decision-makers, as well as the consensus reached about the acceptable risk level. The model does not incorporate technological, competitive, or administrative considerations at the strategic level and so its results should be regarded as merely a guide for decision making and not a definite solution. The best portfolio is obtained when both quantitative and qualitative factors are taken into account. To begin working towards this goal, we develop a zero-one integer programming model as follows. Notice that the optimal solution obtained from this model is only the starting point for the group of decision makers.

Let the column vector $\mathbf{x} = [x_1,\ldots,x_s]$ be a set of zero-one integer variables indicating whether or not project k is included into the portfolio, where s indicates portfolio size (total number of projects available): $x_k=1$ indicates project k is selected and $x_k=0$ indicates project k is not selected. Let the row vector $\mathbf{q} = [q_1,\ldots,q_s]$ be the performance estimates of the project portfolio as indicated by their Internal Rate of Return (IRR). Note that IRR as a performance measure is only suitable if the projects involve roughly similar investments. If the investment amounts differ significantly, a better performance measure might be NPV.

Denote the time and cost dimensions of the projects using the row vectors $\mathbf{t} = [t_1,\ldots,t_s]$ and $\mathbf{c} = [c_1,\ldots,c_s]$, where t_k and c_k, are the completion time and total cost of project k. Also, let $\mathbf{r} = [r_1,\ldots,r_s]$ be the risk vector, where $0\leq r_k \leq 1$ is the risk of project k given as a fraction. Denote the absolute variability associated with the time, cost, and performance dimensions using vectors $\Delta\mathbf{t} = [\Delta t_1,\ldots, \Delta t_s]$, $\Delta\mathbf{c} = [\Delta c_1,\ldots, \Delta c_s]$, and $\Delta\mathbf{q} = [\Delta q_1,\ldots, \Delta q_s]$, where Δt_k, Δc_k, and Δq_k are the absolute deviation of the time, cost, and performance estimates of project k so that $t_k-\Delta t_k \leq t_k \leq t_k+\Delta t_k$, $c_k-\Delta c_k \leq c_k \leq c_k+\Delta c_k$, and $q_k-\Delta q_k \leq q_k \leq q_k+\Delta q_k$. These are assumed to be symmetrical about their mean values. Let B and H be the portfolio's budget and planning horizon, and R be the decision-makers' risk preference. Risk preference is the risk level (in percentage points between 0 and 100%) at which decision makers are comfortable. The set S_m

indicates all mandatory projects. The set P_j indicates all the projects i preceding project j. The set M_j indicates all the projects i that are mutually exclusive with respect to project j. The solution vector is denoted as the column vector $\mathbf{x}^* = [x_1^*,\ldots,x_s^*]$ where x_k^* is the optimal solution for project k indicating whether or not such project should be included in the portfolio. The relative importance of time, cost, and performance are indicated using weight factors denoted as w_t, w_c, and w_q, respectively, where $w_t+w_c+w_q=1$.

MAXIMIZE

$$\mathbf{qx} \tag{85}$$

SUBJECT TO:

a) cost constraint: $$\mathbf{cx} \leq B \tag{86}$$

b) time constraint: $$(\mathbf{t}\text{-}H)\mathbf{x} \leq 0 \tag{87}$$

c) risk constraint: $$(\mathbf{r}\text{-}R)\mathbf{x} \leq 0 \tag{88}$$

d) mandatory: $$x_i = 1 \ \forall \ i \in S_m \tag{89}$$

e) mutually inclusive: $$x_i \geq x_j \ \forall \ i \in P_j \tag{90}$$

f) mutually exclusive: $$x_i + x_j \leq 1 \ \forall \ i \in M_j \tag{91}$$

$$\mathbf{x} = 0,1 \tag{92}$$

Equation (85) is the objective function, equations (86) to (91) constitute the constraints and equation (92) is the technical constraint of the zero-one integer programming formulation for our project selection model. Note that additional constraint equations could be added depending on the situation (e.g., allowing projects to start at different times). The objective function given in equation (85) makes intuitive sense, since it simply states that the total IRR of projects selected should be maximized. Also, the so-called technical constraint in equation (92) simply indicates that the decision variables in the decision vector **x** are binary (zero-one) variables.

However, understanding the constraints given in equations (86) to (91) and the relationship between the objective function and some of these constraints is not as straightforward. To illustrate, we discuss, step by step, a small example.

4.2.1 Marginal Cost

Consider a portfolio of three projects (s=3). The costs are $\mathbf{c}\pm\Delta\mathbf{c}$ = \$ {2000, 1500, 2500} ± {500, 1000, 500}. The total budget is B = \$4,500. Performance as measured by the project's IRR is the only variable that can be part of the objective function. IRR figures are denoted in the performance vectors $\mathbf{q}\pm\Delta\mathbf{q}$. For this example, $\mathbf{q}\pm\Delta\mathbf{q}$ = {8, 7, 5} ± {2, 3, 4} %.

What are all the possible solutions? There are 8 combinations for the solution vector **x**. Selecting no projects or only one project is not wise, because not all the money would be allocated. Selecting all projects, although desirable, is not possible (\$2000+1500+2500 = \$6000 is greater than \$4500). So it seems two projects should be selected, but which two? All two-project combinations are feasible. For **x** = [1,1,0], \$2,000+\$1,500=\$3,500 ≤ \$4,500; for **x** = [1,0,1], \$2,000+\$2,500=\$4,500 ≤ \$4,500; and for **x** = [0,1,1], \$1,500+\$2,500=\$4,000 ≤ \$4,500. To decide, we need to rely upon the concept of marginal contribution. Marginal cost is a measure of how much each percentage point in the project's IRR costs. For project 1, the expected IRR is 8%. Achieving such rate costs \$2000, so each percentage point costs \$2000/8 = \$250. The marginal costs for projects 2 and 3 are \$1500/7 = \$214.29 and \$2500/5 = \$500. Projects 1 and 2 should be included for having the lowest costs for each percentage point. In this case, the optimal solution is $\mathbf{x}^*$ = [1,1,0], assuming equations (85) and (86) are the only ones in the model.

Notice that the uncertainty associated to cost is not included in equation (86). This is because cost uncertainty means that the cost for project l can be as low as c_l-Δc_l and as high as

$c_l+\Delta c_l$, but on average, the cost should be around c_l. As explained before, uncertainty is related to risk, and will be discussed later on.

4.2.2 Planning Horizon

Including time in the portfolio calculations is not as straightforward. Although time is not an additive variable, it can be included in the model if we consider that the average completion time of the selected projects should be less than or equal to a given target. This time target is the planning horizon (H). It is assumed that: a) projects start at time zero, and b) if there are predecessors of any project, all projects with precedence relationships would be considered a single project including all predecessors in it. The average completion time of the projects selected ($\bar{t}$) can be calculated dividing the sum of all expected completion times ($\Sigma x_k t_k$) by the total number of projects selected (Σx_k). Equation (87a) portrays such time constraint. After some algebraic manipulation, equation (87a) is transformed into equation (87b), which is expressed in vector notation in equation (87).

$$\frac{\sum_{k=1}^{s} x_k t_k}{\sum_{k=1}^{s} x_k} \leq H \qquad (87a)$$

$$\sum_{k=1}^{s} x_k \left(t_k - H\right) \leq 0 \qquad (87b)$$

For our example, **t** = {7±4, 3±3, 10±4}. The portfolio's planning horizon is H=14. The objective function remains the same: to maximize the portfolio's IRR as indicated in equation (85). Also, the cost constraint in equation (86) does not change. But the time constraint should be satisfied as well. The same analysis applied to cost can be used here for average time. If all projects are selected, $\bar{t}$ = (7+3+10)/3 = 20/3 = 6.67, which is less than 14. Therefore, the time constraint in equation (87) is not binding and so no further analysis is required. If that was not the case, a combination of both marginal cost and marginal time ought to be considered, choosing the combination that satisfies both constraints while providing the maximum IRR.

4.2.3 Risk Profile

Calculating the portfolio's risk profile is particularly cumbersome. In any case, how can we measure risk? Although risk and uncertainty are not the same, uncertainty can be used as a measure of risk. (We assume risk to be the 'known' unknowns as discussed in the introduction.) Consider our example and the uncertainties for time, cost, and quality. The performance for project 1 can be as high as 8+2=10% or as low as 8-2=6%. The cost can be as high as \$2000+500=\$2500 if particularly unfavorable events arise or as low as \$2000-500=\$1500 in a favorable situation. If project 1 ends up being as costly as possible (\$2500) while achieving its lowest performance (6%), would it be selected as part of the optimal portfolio? Even if project 1 costs \$2500 as opposed to \$2000, the budget limit should not be exceeded (\$2500+\$1500=\$4000 ≤ \$4500). So again, it is a question of marginal cost. Each performance point now would cost \$2500/6 = \$416.67, which is still less than the marginal cost for project 3 (\$500). But if project 3 performs particularly well (5+4=9%) while at the same time being particularly cost effective (\$2500-500=\$2000), then its marginal cost would be \$2000/9 = \$222.22. The latter would change the optimal solution, from **x*** = [1,1,0] to **x*** = [0,1,1].

How likely is the above to happen? In other words, how risky is project 1? It seems that the risk of project 1 depends on how much uncertainty for time, cost, and performance exists for project 1 as well as the combined uncertainty of projects 2 and 3. This requires some form of weighting among dimensions. For example, how much more important is time when compared to cost or specifications? The Analytic Hierarchy Process (AHP), developed by Saaty [70] can be used to assign relative weights based on a series of pairwise comparisons. (AHP is discussed in chapter 5.) Continuing our example, assume that this weighting has been determined to be w_t = 0.35, w_c = 0.40, and w_q = 0.25. These weights are calculated according to the Delta Analytic Hierarchy Process (ΔAHP), which is explained in the next chapter. A measure of relative uncertainty is the uncertainty to average ratio (g_k, k=1,...,s). Each project has three such ratios for time, cost, and performance. The uncertainty to average ratio for time is the uncertainty associated with time divided by the time estimate itself $\Delta t_k/t_k$. The same applies to cost and performance: $\Delta c_k/c_k$ and $\Delta q_k/q_k$. The overall uncertainty to average ratio for project k is the weighted average, $g_k = w_t(\Delta t_k/t_k) + w_c(\Delta c_k/c_k) + w_p(\Delta q_k/q_k)$ $\forall$ k = 1,...,s. In our example, g_1 = 0.3625, g_2 = 0.7238, and g_3 = 0.4200. But as we have seen in the previous paragraph, the risk for project 1 is not only a function of the uncertainty associated with project 1, but a function of the overall uncertainty associated with all the projects. In short, the risk for project k, r_k = $(g_k/(\Sigma g_j))$x100%, j=1,...,s, and $0 \leq r_k \leq 1$. Thus, $r_1 = g_1/(g_1+g_2+g_3)$ = 24.07%, $r_2 = g_2/(g_1+g_2+g_3)$ = 48.05%, and $r_3 = g_3/(g_1+g_2+g_3)$ = 27.88%.

This result is also intuitively sound. Project 1 has the lowest risk, which is consistent with the fact that for project 1 to be unselected and project 3 to be selected, the maximum uncertainty for the extreme cases is required, which is unlikely to happen in reality. Therefore, project 1 is a low risk project. More or less the same is true for project 3. On the other hand, project 2 is highly volatile in its time and cost estimates. Actually, it has the same uncertainty than the combined uncertainty of projects 1 and 3, making project 2 a risky venture. Denote $\bar{r}$ as the average risk of all selected projects. Then, the average risk of the portfolio, $\bar{r}=\Sigma x_k r_k/\Sigma x_k$, should be less than or equal to the maximum risk allowed, R. Equation (88a) depicts such risk constraint.

$$\frac{\sum_{k=1}^{s} x_k r_k}{\sum_{k=1}^{s} x_k} \leq R \tag{88a}$$

Equation (88a) can be transformed into equation (88b) by algebraic manipulation.

$$\sum_{i=1}^{n} x_i (r_i - R) \leq 0 \tag{88b}$$

Equation (88b), which is our risk constraint, is expressed using vector notation in equation (88). What does equation (88b) have to do with risk profile? Risk is a variable between 0 and 1. Chapman and Ward [16] use a risk grid to classify a project as low-risk, medium-risk or high-risk. Since $0 \leq r_k \leq 1$ and given the fact that the squares in the grid are of equal length, a low risk project is a project such that $0 \leq r_k \leq {}^1/_3$, a medium risk project is a project for which ${}^1/_3 < r_k \leq {}^2/_3$, whereas a high risk project has ${}^2/_3 < r_k \leq 1$. A low-risk, medium-risk, or high-risk portfolio would have values for R of ${}^1/_3 = 33{}^1/_3\%$ (low-risk profile), ${}^2/_3 = 66{}^2/_3\%$ (medium-risk profile), and ${}^3/_3 = 100\%$ (high-risk profile), respectively.

4.2.4 Mandatory, Mutually Inclusive and Mutually Exclusive Projects

So far we have not taken into account the relationships among projects and their constraints (mandatory, mutually inclusive, and mutually exclusive projects). Mutually inclusive projects are projects with precedence relationships and can in fact be considered a single project. Projects can be mutually exclusive (either one or the other is selected, but not both) in case of competing technologies or strategic alternatives. Mandatory projects are projects that must be selected in order to satisfy given technological, corporate, or industry requirements. Equation (89) forces mandatory projects to be selected because if project i is a mandatory project (i.e. project i is in set S_m), the equation x_i=1 forces x_i to be one. The mutually inclusive constraint is equation (90). If project i precedes project j, then $x_i \geq x_j$, so that project i must be selected if project j is selected, and project j cannot be selected if project i is not selected. The mutually exclusive constraint is given by equation (91) because at most only one of the variables is allowed to equal one.

Following our example let projects 1 (Alpha) and 3 (Gamma) be mutually exclusive: they cannot be selected at the same time. This means that the solution **x** = [1,0,1] is not feasible. (In any case, such solution would also break the time constraint.) The other two feasible solutions that would make better sense are **x** = [1,1,0] and **x** = [0,1,1]. But project 2 (Beta) requires project 3 (Gamma) as a predecessor. (Gamma precedes Beta.) In both feasible solutions, Beta is included, but only for **x** = [0,1,1] is Gamma also selected. Although selecting Beta and Gamma as opposed to selecting Alpha and Beta would not yield the highest performance ($7+5=12 < 8+7=15$), such choice would be the only one satisfying the constraint for mutually exclusive projects. In our example, when project relationships are considered, the priority/risk tradeoff is not binding. Only one optimal solution exists: to select project 2 (Beta) and project 3 (Gamma), because although project 1 (Alpha) would be the preferred choice for low-risk investors, the only feasible way to include Alpha is by excluding Gamma, which automatically implies that Beta cannot be included due to the Gamma→Beta (or 3→2) precedence relationship. Therefore, the optimal (and feasible) portfolio is the high-risk portfolio: to select Beta and Gamma. Therefore, when these dependency constraints are considered, the optimal solution changes from **x*** = [1,1,0] to **x*** = [0,1,1].

Figure 14 shows the integer linear programming formulation for the example discussed so far assuming that the decision-makers' risk profile is medium ($\overline{R} \leq 66^2/_3\% \approx 67\%$). Since (32+92+39)/3 = 54 ≤ 67%, risk is not a binding constraint, so the optimal solution does not change.

Maximize $8\ x_1 + 7\ x_2 + 5\ x_3$

Subject to

a) cost constraint: $2000\ x_1 + 1500\ x_2 + 2500\ x_3 \leq 4500$

b) time constraint: $-7\ x_1 - 11\ x_2 - 4\ x_3 \leq 0$

c) risk constraint: $-\ 35\ x_1 + 25\ x_2 - 28\ x_3 \leq 0$

d) mutually inclusive: $x_1 + x_3 \leq 1$

e) mutually exclusive: $x_2 - x_3 \leq 0$

$x_1, x_2, x_3 = 0,1$

Figure 14. Integer Linear Programming Model.

4.3 Ranking Projects to Assist Managers in Prioritizing Projects

As we can see, the 0-1 integer programming model maximizes performance while keeping cost, time, and risk in check. Generally speaking, we can say that good projects consistently have relatively high performance and relatively low time and cost figures. Consider our example. The best (lowest) time is for project 2, followed by project 1 and finally by project 3. The best (lowest) cost is for project 2, followed by project 1 and finally by project 3. The best performance (highest figure) is for project 1, closely followed by project 2 and with project 3 last. It seems the best project to select is project 2 (2 out of 3 best figures), followed by project 1 (1 out of 3). Project 3 is certainly not a wise choice. However, from the zero-one integer programming model we have that project 1 should be left out of the solution, so that project 3 is in second place after project 2. What have we just done? We have selected a set of projects for portfolio. Among these selected projects we can now formalize a priority index and use it to generate a list sorted by rank in order to classify projects as high priority, medium priority, and low priority. This will help decision makers to make further allocations among projects in the portfolio.

The priority index summarizes all the estimates and their relative priority when compared with the portfolio (set of available projects). Let Z_k be the classification index for project k, where $0 \leq Z_k \leq 1 \ \forall \ k=1,\dots,s$. A project is considered to be low priority if $0 \leq Z_k \leq {}^1/_3$, medium priority if ${}^1/_3 < Z_k \leq {}^2/_3$, and high priority if ${}^2/_3 < Z_k \leq 1$. This scheme implicitly assumes that the higher the index, the better the project. The classification index must include the three project management dimensions. The weights for time, cost, and performance (w_t, w_c, and w_q) can be used to embody the three dimensions into one single number. But the units for time, cost and performance are not equivalent (we have time units such as years, money units such as thousands of dollars, and performance units in percentage). So first, we have to transform these figures into ratios. Let t_{Min} and t_{Max} be the minimum and maximum time estimates ($t_{Min} = Min\{t_k\}$and $t_{Max} = Max\{t_k\} \forall$ $k=1,\dots,s$), c_{Min} and c_{Max} be the minimum and maximum cost estimates ($c_{Min} = Min\{c_k\}$ and $c_{Max} = Max\{c_k\}$, $k=1,\dots,s$), and q_{Max} be the maximum performance estimate ($q_{Max} = Max\{q_k\} \forall$ $k=1,\dots,s$). Consider performance first. The performance ratio (rq_k) for project k is q_k/q_{Max}.

Let rt_k and rc_k be the time and cost ratios for project k, k = 1,...,s. Can we follow the same reasoning for time and cost? The answer is no, because the best projects have the lowest time and cost estimates. But the classification index sorts projects from the highest to the lowest ratio. So we need to invert the time and cost figures, assigning the lowest figure to the highest estimate and viceversa. In order to do that, we have to subtract from the maximum value of all the estimates the given estimate for each project (t_{Max}-t_i and c_{Max}-c_i). The best estimate is always going to be t_{Max}-t_{Min}. Thus, $rt_k = (t_{Max}-t_k)/(t_{Max}-t_{Min})$. A similar reasoning can be applied to cost: $rc_k = (c_{Max}-c_k)/(c_{Max}-c_{Min})$. The overall index is the weighted average of these ratios, $Z_k = w_t rt_k + w_c rc_k + w_q rq_k$, k=1,...,s as indicated in equation (93).

$$Z_k = w_t \frac{\text{Max}\{t_j\} - t_k}{\text{Max}\{t_j\} - \text{Min}\{t_j\}} + w_c \frac{\text{Max}\{c_j\} - c_k}{\text{Max}\{c_j\} - \text{Min}\{c_j\}} + w_q \frac{q_k}{\text{Max}\{q_j\}}, j=1,\ldots,s \tag{93}$$

In our example, $Z_1 = 0.6$, $Z_2 = 0.96875$, and $Z_3 = 0.15625$. This means that project 2 has higher priority than project 1, which in turn has a higher priority than project 3. In short, $x_2>x_1>x_3$. Actually, project 2 is a high priority project ($^2/_3<0.97\leq1$), project 1 is a medium priority project ($^1/_3<0.60\leq^2/_3$), and project 3 is a low priority project ($0\leq0.16\leq^1/_3$). For decision-makers this means that in principle they should select project 2 first, then select (if possible) project 1 (which according to the results from the zero-one integer programming model should not be selected) and finally select project 3 if all constraints are satisfied. But this is not necessarily the best decision, because equation (93) does not consider the risk profile. For example, project 2 scores the highest, but at the same time is the riskiest. Although project 1 scores not as high, its risk is the lowest of all. Our attempt is to help decision-makers to quickly realize which projects should be left out for sure by interacting with data and choices through a

GDSS. This priority index could be a useful adjunct to the ILP model developed in the previous section.

4.4 Discussion

Ultimately, the portfolio is a tradeoff decision between priority and individual risk preferences. In our example, project 2 would be the preferred choice for high-risk investors while project 1 would be the best choice for low-risk investors. Given the fact that decisions are going to be taken by a group of people, there is no such thing as an optimal solution, and some form of consensus or at least compromise will be required.

To apply the mathematical model for portfolio selection from among a realistically large set of projects discussed in this chapter requires more than the system of equations proposed and the simple priority index calculations; a Group Decision Support System (GDSS) is required. In a GDSS, the mathematical model is one of the components; another important component is the interface between the end user and the computer. Such interface should allow users to consider the solution obtained from the mathematical model and make any changes desired based on empirical evidence, experience, intuition, or strategic considerations. Consider, for example, the integrated framework for project portfolio selection discussed by Archer and Ghasemzadeh [2].

Although the model finds the optimal portfolio, sometimes the optimal is not the best option given managerial considerations. Therefore, it is important to portray the three project management dimensions plus risk in such way that makes intuitive sense to the user. This would in turn allow users to make changes to the optimal solution proposed in light of strategic or technological considerations beyond the scope of our mathematical model, which requires considering the interface used to portray portfolio data. Taking into account the preferences of decision makers that are difficult to quantify is the topic of discussion in the next chapter.

Chapter 5. The Delta Analytic Hierarchy Process (ΔAHP)

5.1 Introduction

The Analytical Hierarchy Process (AHP), developed by Saaty [70] is a multicriteria decision-making theory for modeling unstructured problems in economic, social and management sciences, based on the fundamental process underlying perception: decomposition and synthesis [42][71]. AHP allows a single or a group of decision-maker(s) to quantitatively analyze a qualitative problem. It has been embedded in an interactive commercial software package called Expert Choice[3]. Furthermore, it provides a systematic framework for the entire decision process, and to validate the consistency of the arguments. The two main advantages of AHP are the accountability for such inconsistencies and the verbalization of the pairwise comparisons. (Pairwise comparisons in AHP are comparisons of each criterion against each other criterion, assigning a number as a judgment of the degree in which the first criterion is more important than the second criterion.) The main disadvantage is the amount of time required to break down the problem, deciding upon which criteria to quantify against the overall objective, and verbalizing the inherent judgments of the decision-makers (compromising when necessary). If the group concludes that a low consistency ratio reflects a poor problem breakdown process, they ought to reconsider the situation [4][8][30][46][75].

[3] http://www.expertchoice.com/

But judgments are not deterministic figures. If the attribute assigned to each criterion is intangible or if enough data are available to compute statistical errors (each decision-maker assigns a different attribute due to individual judgment differences but nonetheless a final judgment is reached), the uncertainty associated with the judgment matrix should be considered. (The judgment matrix is the matrix in which all the pairwise comparisons are entered.) The Stochastic Analytic Hierarchy Process (ΔAHP) is a stochastic methodology because it portrays the judgments as probabilistic figures (each pairwise comparison in the judgment matrix has an associated variation in the error matrix). In this chapter, we describe the development of a prototype system that implements AHP using estimated uncertainties in the data.

5.2 Delta AHP

Rosenbloom [69] recommends portraying the pairwise comparisons as random variables, varying from 1/9 to 9, and calculating probabilistic weights using simulation. Zahir [91] explores the situation in which the fuzziness of each judgment is portrayed as a percentage above and/or below the final judgment reached. Such uncertainty may be derived from the error of data measurements and/or the confidence of intangible criteria. This discussion focuses on the algorithm developed by Zahir. This was used to model, build and test a prototype software system named Delta AHP (ΔAHP). The development efforts were based on the synthesis or black-box problem [87], in which little is known about the nature of the system. Nevertheless, the excitation (input) and the response (output) are known from data available in the examples used in Zahir's paper. The ΔAHP methodology is illustrated in Figure 15, which is based on the decision-support model of Simon [77].

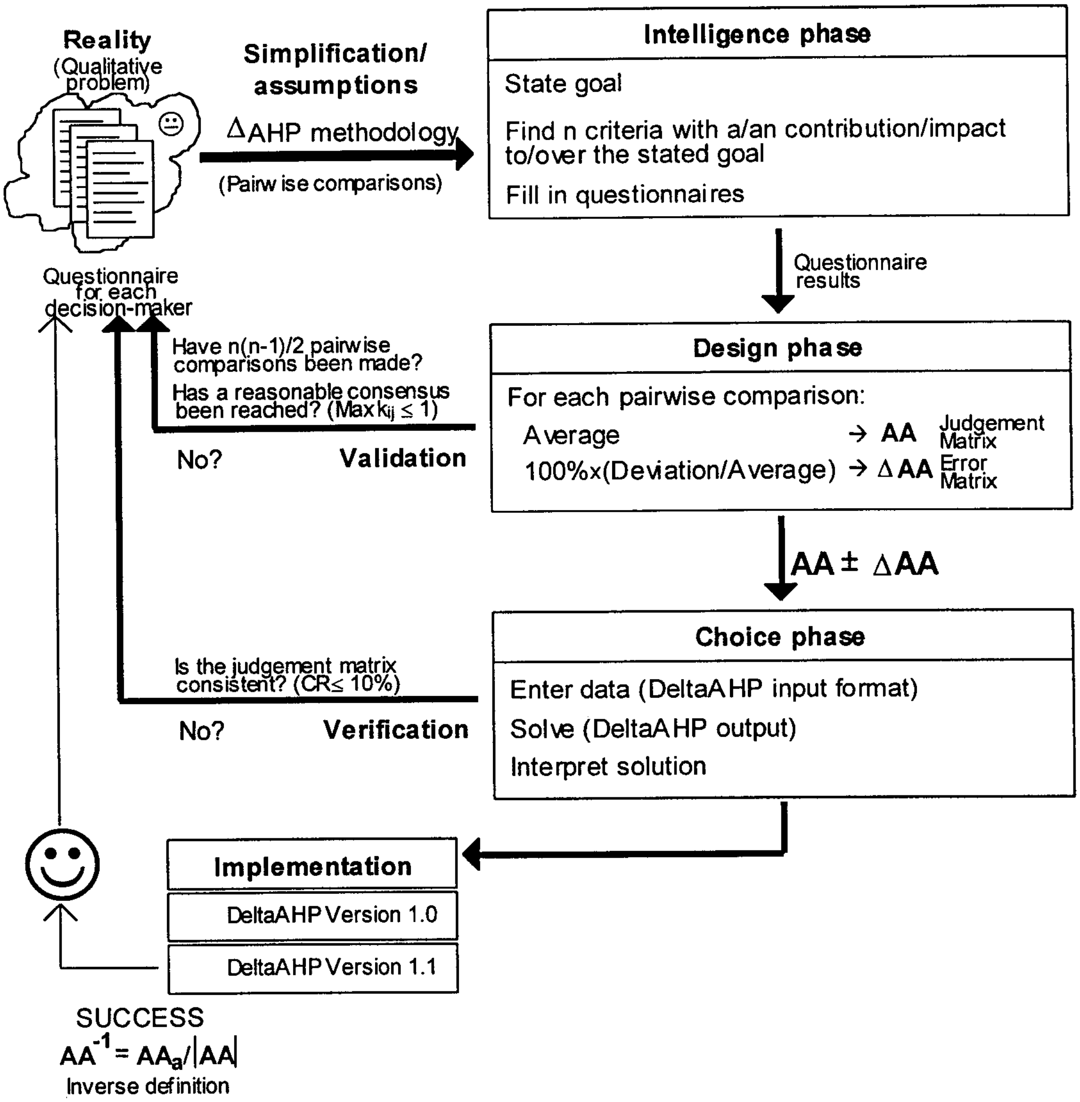

Reality
(Qualitative problem)
Simplification/ assumptions
ΔAHP methodology
(Pairwise comparisons)
Intelligence phase
State goal
Find n criteria with a/an contribution/impact to/over the stated goal
Fill in questionnaires
Questionnaire results
Questionnaire for each decision-maker
Have n(n-1)/2 pairwise comparisons been made?
Has a reasonable consensus been reached? (Max kij ≤ 1)
No?
Validation
Design phase
For each pairwise comparison:
Average → AA Judgement Matrix
100%x(Deviation/Average) → ΔAA Error Matrix
AA ± ΔAA
Is the judgement matrix consistent? (CR≤ 10%)
No?
Verification
Choice phase
Enter data (DeltaAHP input format)
Solve (DeltaAHP output)
Interpret solution
Implementation
DeltaAHP Version 1.0
DeltaAHP Version 1.1
SUCCESS
AA⁻¹ = AAa/|AA|
Inverse definition

Figure 15. The ΔAHP Development Process.
Adapted from Turban and Aronson [84].

The intelligence, design, and choice phases illustrate how to apply ΔAHP to the real-life problem (qualitative in nature) faced by a single or a group of ζ decision-maker(s). ΔAHP uses a problem breakdown process to analyze the situation. First (intelligence phase), decision-makers state the goal (most relevant aspect of the problem under study that calls for analysis) and a set of n criteria with contribution/impact to/over the stated goal are defined. For each combination of two of these criteria, a quantitative pairwise comparison is made by each decision-maker. Second (design phase), ζ judgment matrices are obtained from the ζ questionnaires filled out by decision-makers. The average and the standard deviation are calculated and the judgment and error matrices are obtained. Third (choice phase), the data are entered into Delta AHP and the results are obtained and discussed. A consistency ratio greater than 10% indicates contradictions in the arguments. In that case, decision-makers should consider the problem again from a different point of view, trying to reach or modify the criteria and/or the quantitative judgments assigned. On the other hand, if the consistency ratio is close enough to 10% and the arguments are sound, no modification is required [69][70][75][91]. The last phase (implementation) deals with the detailed instrumentation of ΔAHP.

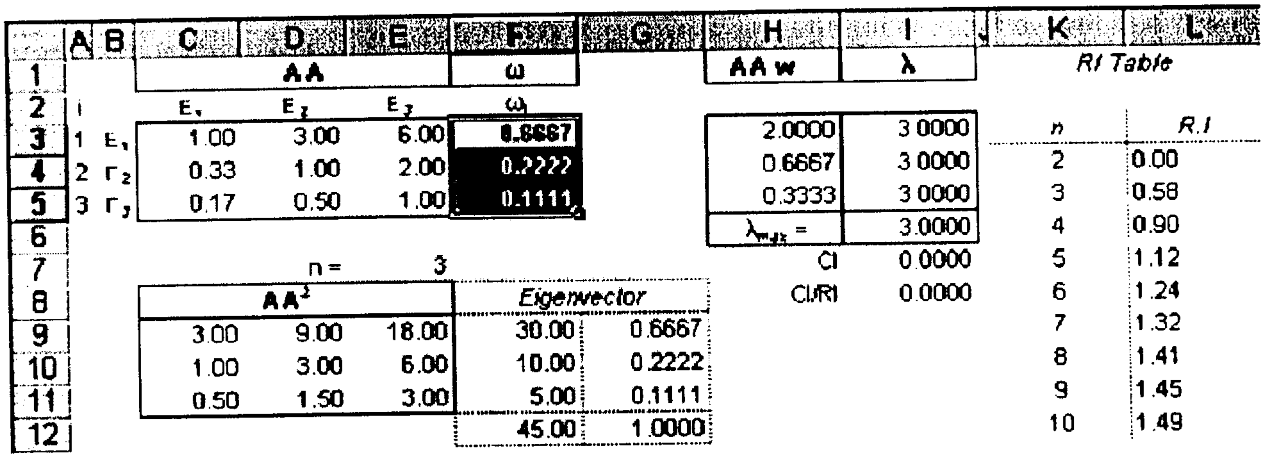
5.3 Methodology

Priority in ΔAHP and in any other circumstance measures the relative intensity of what is important to people (decision-makers). The Stochastic Analytic Hierarchy Process (ΔAHP) is an extension of Saaty's Analytic Hierarchy Process (AHP). The underlying ratio scale and the semantic scale used are the same in both cases. Implicit (as to the relative part) is the idea of comparisons among elements. ΔAHP is based on matrix theory and ratios. A ratio (or percentage) is a number between 0 and 1 used to portray relative importance (weight). The problem breakdown process allows decision-makers to define a set of criteria that have a contribution/impact to/over the stated goal. The problem is how to calculate such ratios (one for each criterion). Let ν be the number of criteria defined, and ω_i be the relative weight of each criterion, where $\Sigma_i \omega_i = 1, \forall i = 1,\ldots,\nu$. If e_{ij} describes the relationship between criterion i and j, then $e_{ij}=\omega_i/\omega_j$. The ω's are unknown. Nonetheless, it is possible to ask decision-makers to verbalize a series of pairwise comparisons and then to assign an attribute to each relevant judgment. Based on empirical evidence, Saaty recommended the use of a 1 to 9 semantic scale for such purpose [70]. Odd numbers are reference points. Even numbers are used for compromising when reaching a final judgment is difficult. Of all the possible ν^2 pairwise comparisons, there are ν pointless comparisons (when i=j the criterion is compared against itself; therefore $e_{ij}=e_{ji}=1$). The remaining $\nu^2 - \nu = \nu(\nu-1)$ comparisons have to be made once only, because if criterion i is e_{ij} times more important than criterion j, then criterion j is $1/e_{ij} = e_{ji}$ times more important than criterion i, which means that only half of these $\nu(\nu-1)$ judgments are required.

5.3.1 Consistency

By consistency we mean here not merely the traditional requirement of the transitivity of preferences (if apples are preferred to oranges and oranges are preferred to bananas, then apples are preferred to bananas), but the actual intensity with which the preference is expressed transits through the sequence of objects in the comparison. For example, if apples are twice as preferable as oranges and oranges are three times as preferable as bananas, then apples must be six times as preferable as bananas. This is what we call cardinal consistency in the strength of preference. Inconsistency is a violation of proportionality, which may or may not entail violation of transitivity. What matters is not whether we are consistent on particular comparisons, but how strongly consistency is violated in the numerical sense for the overall problem under study. [72]

Figure 16 shows the computations of the eigenvector (ω) for a perfectly consistent judgment matrix (**AA**) using Saaty's successive squaring algorithm ($\mathbf{AA}^2$,$[\mathbf{AA}^2]^2$=$\mathbf{AA}^4$,$[\mathbf{AA}^4]^2$= $\mathbf{AA}^8$, and so on). Since **AA** is perfectly consistent, only one iteration ($\mathbf{AA}^2$) is required. The example is particularly relevant because it allows us to further explore and grasp the concept of consistency.



	A	B	C	D	E	F	G	H	I	J	K	L
1				AA		ω		AA w	λ		*RI Table*	
2	i		E_1	E_2	E_3	ω_i						
3	1	E_1	1.00	3.00	6.00	0.6667		2.0000	3.0000		*n*	*R.I*
4	2	E_2	0.33	1.00	2.00	0.2222		0.6667	3.0000		2	0.00
5	3	E_3	0.17	0.50	1.00	0.1111		0.3333	3.0000		3	0.58
6								λ_{max} =	3.0000		4	0.90
7				n =	3			CI	0.0000		5	1.12
8				AA^2		*Eigenvector*		CI/RI	0.0000		6	1.24
9			3.00	9.00	18.00	30.00	0.6667				7	1.32
10			1.00	3.00	6.00	10.00	0.2222				8	1.41
11			0.50	1.50	3.00	5.00	0.1111				9	1.45
12						45.00	1.0000				10	1.49

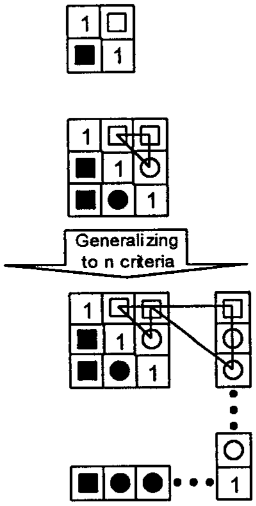
Figure 16. Deterministic AHP by successive squaring.

In the judgment matrix, criterion E_1 is 3 times more important than criterion E_2, and E_2 is 2 times more important than E_3. If the semantic scale (all the e_{ij}'s in the judgment matrix) is as good a scale as the underlying ratio scale (the ω's obtained as described a few lines above), then it should conform to the same properties the ratio scale does. In a perfectly consistent matrix, using e_{ij}'s (the semantic scale) or ω_i's (underlying ratio scale) is equivalent. Therefore, to say that E_1 is 3 times (e_{12}) more important than E_2 is equivalent to saying that ω_1 is 3 (e_{12}) times more important than ω_2. Likewise, ω_2 is e_{23} times more important than ω_3. Since $\omega_1+\omega_2+\omega_3 = 1$, the relationship between E_1 and E_3 is constrained by this system of 3 equations if perfect consistency is to be assured. Substituting $\omega_2=e_{23}\omega_3$ into $\omega_1=e_{12}\omega_2$, we obtain the relationship between E_1 and E_3 for perfect consistency as $\omega_1=e_{12}(e_{23}\omega_3)=e_{13}\omega_3$. Thus, $e_{13}=e_{12}e_{23}$. Also, notice that ν-1=3-1=2 independent pairwise comparisons were entered (e_{12}=3, and e_{23}= 2, being $e_{13}=e_{12}e_{23}$=3x2=6). Generalizing to ν criteria, if out of the $\nu(\nu-1)/2$ pairwise comparisons required, only ν(ν-1)/2-(ν-1)=(ν-1)(ν-2)/2 pairwise comparisons that follow the consistency equation (94) are entered, then the judgment matrix is perfectly consistent.

$$e_{ij} = \frac{e_{ik}}{e_{jk}} \tag{94}$$

When **AA** is consistent, a set of ν interconnected judgments of the form $e_{i,i1}, e_{i1,i2}, e_{i2,i3}, \ldots, e_{n-1,j}$ can be formed across the rows and columns. Based on this abstract construct, it is possible to build perfectly consistent matrices. To do so, the relationship $e_{ij}e_{jk}=e_{ik}$ (the same as saying that element e_{ij} equals e_{ik}/e_{jk}, where e_{ij} is dependent on the two remaining elements on the k^th^ column and the i^th^ or j^th^ rows of the triangular rectangle formed), has to be kept. We have shown that out of the ν(ν-1)/2 pairwise comparisons decision-makers have to decide upon (ν comparisons have a value of one since they are self comparisons, and the remaining ν^2-ν-ν(ν-1)/2=ν(ν-1)/2 judgments are the reciprocals of the original ν(ν-1)/2 subjective appraisals), only ν-1 are strictly required if the remaining ν(ν-1)/2-(ν-1)=(ν-1)(ν-2)/2 judgments are calculated using the consistency relationship $e_{ij}=e_{ik}/e_{jk}$ described in equation (94) (see Figure 17).

If more than ν-1 judgments are made which do not follow the consistency equation, then the consistency ratio can be used as a statistical test. A randomly generated matrix is a matrix of nonsense (random) judgments. As long as **AA** is 10% or less close to such nonsense matrix, the decision process followed to arrive to the judgments is considered coherent enough because these judgments do not result in a totally interdependent measure. To artificially ensure perfect consistency is not a good idea at all, because an advantage of AHP is precisely the fact that it provides a measure of consistency. If all the ν(ν-1)/2 pairwise comparisons are requested from decision-makers and if the consistency ratio is less than 0.10 but greater than 0.0 (perfect consistency), it is reasonable to consider that the judgments and the results obtained from them are sound [72]. Low inconsistency is necessary but not sufficient for a good decision. It is possible to be perfectly consistent as we have seen, yet consistently wrong. Nevertheless, perfect consistency is important in our discussion because of its implications when the stochastic AHP algorithm is used, as we will see.



For the simplest case ($\nu = 2$), it is clear the equation derived above holds (ν-1=1 element has to be entered). The ones in the main diagonal can not affect any other e_{ij} ($i \neq j$). Thus, the remaining $\nu^2 - \nu = 4 - 2 = 2$ numbers have to be entered. Since half of these are the reciprocal of the remaining half, only 2/2 = 1 judgement is required.

For $\nu = 3$, the former situation holds (i.e., the reasoning applicable to the previous $\nu = 2$ submatrix is the same). Nevertheless, if we allow the decision-maker to enter any judgement for the first row, third column, the number in row 2, column 3 can be calculated using the other two non-unity elements ($e_{23} = e_{21}/e_{31}$, for example). Certainly, the same would be true if the judgement in row 2, column 3 would be entered. In that case, the remaining element could be calculated to assure consistency.

□ Judgement should be entered by decision-makers
■ Judgement aji is the reciprocal of judgement aij
○ Judgement can always be derived based on previous judgements
● Recriprocal of the calculated judgement

Figure 17. Ensuring Perfect Consistency.

5.3.2 Judgment Uncertainty

Let us now further expand the paradigm by allowing uncertainty in the quantitative measurements (AHP→ΔAHP). This uncertainty is visualized as a sort of fuzziness around the semantic scale: $\omega_i \rightarrow \omega_i \pm \Delta\omega_i$ and $\omega_j \rightarrow \omega_j \pm \Delta\omega_j$. For simplicity, the uncertainty is assumed to be symmetric. The uncertainty associated with the judgment matrix **AA** is entered in the uncertainty matrix **ΔAA**, where **ΔAA** = Δe_{ij} $\forall$ i=1,…,n, j=1,…,n. Then, $e_{ij} \rightarrow e_{ij} \pm \Delta e_{ij} \rightarrow (\omega_i \pm \Delta\omega_i)/(\omega_j \pm \Delta\omega_j) \approx \omega_i/\omega_j(1 \pm \delta_{ij}) = \omega_{ij}(1 \pm \delta_{ij})$, where $\delta_{ij} = \Delta\omega_i/\omega_i + \Delta\omega_j/\omega_j$ and $\Delta e_{ij} = \omega_{ij}\delta_{ij}$. The approximation is suitable if $\Delta\omega_i \ll \omega_i$ and $\Delta\omega_j \ll \omega_j$. **ΔAA** expresses the confidence level of the decision-maker over each judgment. Since having absolute numbers when portraying such variation is difficult for the human mind (particularly given the structure of the judgment matrix, i.e., assigning Δe_{ij}'s to each e_{ij}, when $e_{ij}=1/e_{ji}$, but $\Delta e_{ij} \neq 1/\Delta e_{ji}$), a 0% to 100% confidence level is used instead. Zahir [91] developed an ingenious algorithm (see Figure 18) to calculate the weights and their uncertainty ($\mathbf{VV} \pm \delta\mathbf{VV}$) based on the information provided by decision-makers ($\mathbf{AA} \pm \Delta\mathbf{AA}$). $\mathbf{VV}_k$ and $\delta\mathbf{VV}_k$ are the unnormalized eigenvector and its uncertainty, respectively.

```
Step 1   $k = 1, \mathbf{VV}_{k-1} \leftarrow (1,..,1)^T, \mathbf{UU}_{k-1} = \mathbf{0}, \mathbf{CC} = \mathbf{0}, \delta\mathbf{VV}_{k-1} = \mathbf{AA}^{-1}\,\Delta\mathbf{AA}\,\mathbf{VV}_{k-1}$.
Step 2   $\mathbf{VV}_k = \mathbf{AA}\,\mathbf{VV}_{k-1}$.
Step 3   $\delta\mathbf{VV}_k = \mathbf{AA}\,\delta\mathbf{VV}_{k-1} + \Delta\mathbf{AA}\,\mathbf{UU}_{k-1}$.
Step 4   $\mathbf{CC}_{[i]} = \mathbf{VV}_{k[i]}/\mathbf{VV}_{k-1[i]}$; for all $i = 1,\ldots,\nu$, where i implies the $i^{\underline{th}}$ element.
Step 5   $CC_0 = |\Sigma_i \mathbf{CC}_{[i]} - \nu\mathbf{CC}_{[1]}|/\nu\mathbf{CC}_{[1]}$; for all $i = 1\ldots\nu$, where i implies the $i^{\underline{th}}$ element.
Step 6   If $CC_0 > \varphi$ (where $\varphi$ is arbitrarily small), then $\mathbf{UU}_k = \mathbf{VV}_k + \delta\mathbf{VV}_k$;
         go to Step 3 ($k = k+1, \mathbf{UU}_{k-1} = \mathbf{UU}_{k-2}, \mathbf{VV}_{k-1} = \mathbf{VV}_{k-2}, \delta\mathbf{VV}_{k-1} = \delta\mathbf{VV}_{k-2}$).
Step 7   Normalize $\mathbf{VV}_k \rightarrow \mathbf{VV}$ and $\delta\mathbf{VV}_k \rightarrow \delta\mathbf{VV}$.
         To normalize $\mathbf{VV}_k$:
         $\mathbf{VV}_{[i]} = \mathbf{VV}_{k[i]}/\Sigma_i\mathbf{VV}_{k[i]}$; for all $i = 1\ldots\nu$, where i implies the $i^{\underline{th}}$ element.
         To normalize $\delta\mathbf{VV}_k$:
         $\delta\mathbf{VV}_{[i]} = |(\mathbf{VV}_{k[i]}+\delta\mathbf{VV}_{k[i]})/\Sigma_i(\mathbf{VV}_{k[i]}+\delta\mathbf{VV}_{k[i]}) - (\mathbf{VV}_{k[i]}-\delta\mathbf{VV}_{k[i]})/\Sigma_i(\mathbf{VV}_{k[i]}-\delta\mathbf{VV}_{k[i]})|/2$.
         $\mathbf{VV}$ is the principal eigenvector and $\delta\mathbf{VV}$ is the vector containing uncertainties
         ($\mathbf{VV} \pm \delta\mathbf{VV}$ founded in k iterations). The largest eigenvalue is $\lambda_{max} = \mathbf{CC}_{[1]}$.
```

Figure 18. Zahir's Algorithm.

5.4 Delta AHP

Delta AHP was developed using Borland Delphi 5. Delta AHP 1.1 uses a Rich Text File (*.RTF) editor, which provides compatibility with other packages. Delta AHP files (*.AHP) can be imported using Microsoft Word and output or input files can be directly copied into Microsoft Excel (select and copy in Delta AHP and then paste in Excel). A predefined sequence for data entry is required (see Figure 19). Deterministic (no uncertainty for **AA**) or probabilistic (both **AA** and ΔAA) data can be entered. Further releases will include the standard AHP algorithm (successive matrix squaring) for deterministic problems.

	Probabilistic input (AA±ΔAA)	Deterministic input (AA)
1	< DeltaAHP FILE >	< AHP FILE >
2	Probabilistic	Deterministic
3	v	v
4	φ	φ
5	< CRITERIA >	< CRITERIA >
6	Criterion 1	Criterion 1
7	Criterion 2	Criterion 2
8	Criterion 3	Criterion 3
5+n	Criterion v	Criterion v
6+n	< PAIRWISE COMPARISON MATRIX (AA) >	< PAIRWISE COMPARISON MATRIX (AA) >
7+n	$1 \quad e_{12} \quad e_{13} \; \ldots \; e_{1n}$	$1 \quad c_{12} \quad c_{13} \; \ldots \; c_{1n}$
8+n	$e_{21} \quad 1 \quad e_{23} \; \ldots \; e_{2n}$	$c_{21} \quad 1 \quad c_{23} \; \ldots \; c_{2n}$
9+n	$e_{31} \quad e_{32} \quad 1 \; \ldots \; e_{3n}$	$c_{31} \quad c_{32} \quad 1 \; \ldots \; c_{3n}$
6+2n	$e_{n1} \quad e_{n2} \quad e_{n3} \; \ldots \; 1$	$c_{n1} \quad c_{n2} \quad c_{n3} \; \ldots \; 1$
7+2n	< UNCERTAINTY MATRIX (Δ AA) AS A PERCENTAGE OF AA>	
8+2n	$0 \quad \Delta e_{12} \quad \Delta e_{13} \; \ldots \; \Delta e_{1n}$	
9+2n	$\Delta e_{21} \quad 0 \quad \Delta e_{23} \; \ldots \; \Delta e_{2n}$	
10+2n	$\Delta e_{31} \quad \Delta e_{32} \quad 0 \; \ldots \; \Delta e_{3n}$	
5+3n	$\Delta e_{n1} \quad \Delta e_{n2} \quad \Delta e_{n3} \; \ldots \; 0$	

Note: When a zero is entered in AA, DeltaAHP looks for the reciprocal ($e_{ij} = 1/e_{ji}$), whereas a zero in ΔAA tells the program to use the same confidence level ($\Delta e_{ij} = \Delta e_{ji}$).

Figure 19. Delta AHP Input Format.

Zahir's seven step algorithm does not depend on the assumption (for analytical purposes) that second order terms are equal to zero for a good approximation. Nevertheless, the inverse of the pairwise comparison matrix ought to be obtained. One of the most efficient ways to do so is by applying the Gauss-Jordan numerical method, which relies on row operations to calculate the inverse matrix. However, due to the nature of the matrix **AA**, this method does not work. Appendix A discusses the problem of the Gauss-Jordan matrix inversion process.

As a result, the inverse of the pairwise comparison matrix (**AA**$^{-1}$) was calculated using the definition of inverse. Although the procedure takes more time, most judgment matrices are relatively small, so speed is not an issue. The inverse matrix **AA**$^{-1}$ equals the adjoint matrix of **AA** (**AA**$_a$) divided by the determinant of **AA** as indicated in equation (95).

$$\mathbf{AA}^{-1} = \mathbf{AA}_a/|\mathbf{AA}| \tag{95}$$

The adjoint matrix of **AA** is the transpose of the matrix of cofactors, $\mathbf{AA_c}$ as shown in equation (96).

$$\mathbf{AA_a} = \mathbf{AA_c'} \tag{96}$$

The nxn matrix of cofactors of **AA** is the matrix for which each element e_{ij} is equal to $(-1)^{i+j}$ times the determinant of the minor of **AA** in row/column i/j as shown in equation (97).

$$\mathbf{AA_c'} = (-1)^{i+j}\,|\mathbf{AA_{m\,i,j}}| \tag{97}$$

The minor of **AA** in i,j ($\mathbf{AA_{m\,i,j}}$) is the (n-1)x(n-1) matrix formed when row i and column j are removed from **AA**. This set of equations requires matrix **AA** not to be singular (the determinant of AA, |AA|, has to be different than zero).

The next question to answer is when a judgment matrix is singular. For a 3x3 judgment matrix, its determinant, $|AA|=-e_{13}+e_{23}e_{12}$, which has to be different than zero for the inverse to exist ($e_{12}e_{23}-e_{13} \neq 0$; $e_{12} \neq e_{13}/e_{23}$). By generalizing this concept, we obtain the "invertibility" equation (98).

$$e_{ij} \neq \frac{e_{ik}}{e_{jk}} \tag{98}$$

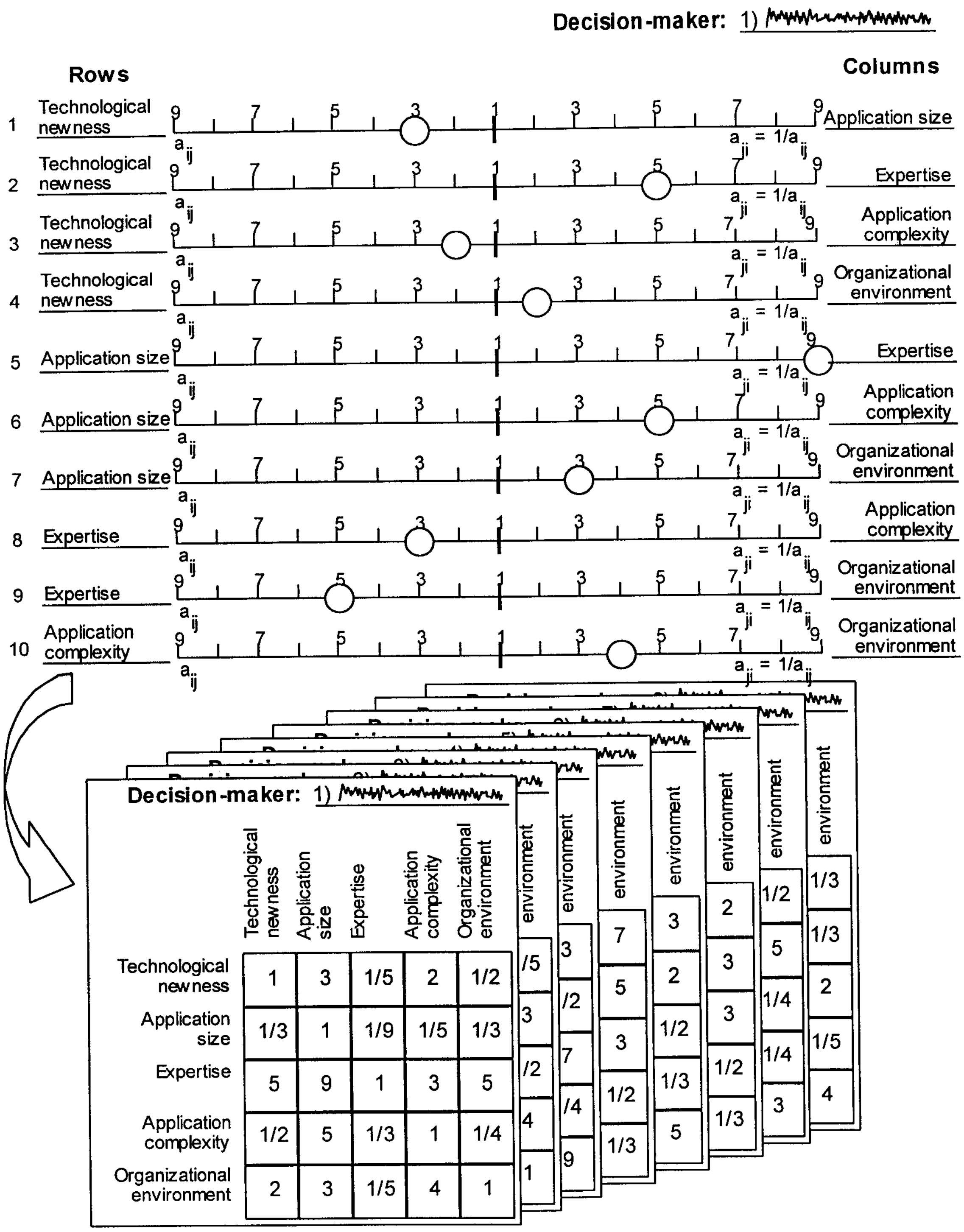
After extensive analysis, a simple procedure for building perfectly consistent matrices was found for experimentation purposes. It is discussed in Appendix B.

Notice that the consistency (94) and the invertibility (98) equations are strictly contradictory. In other words, the inverse of perfectly consistent matrices does not exist because the related system of equations is under-determined. Therefore, ΔAHP requires the judgment matrix not to be perfectly consistent.

5.5 System Implementation

Although the inverse of the judgment matrix is not required for standard AHP computations (successively squaring **AA**), it is interesting the fact that the impossibility of calculating the inverse of perfectly consistent matrices has not been analyzed by Saaty. In our discussion, we highlighted this fact by exploring what consistency means in practice using relatively small examples.

The seven-step algorithm presented here (see Figure 18) has some modifications and additions compared to Zahir's work. (The final weights and their associated uncertainty are normalized, φ is an input parameter, among other adaptations.) The results obtained from Delta AHP were compared with the output from Zahir's original paper for validation purposes. Normalizing the uncertainty vector requires upper and lower bounds for the absolute value of the eigenvector. The normalized uncertainty above and below each normalized ΔAHP weight is half the absolute difference.



Decision-maker: 1)

	Technological newness	Application size	Expertise	Application complexity	Organizational environment
Technological newness	1	3	1/5	2	1/2
Application size	1/3	1	1/9	1/5	1/3
Expertise	5	9	1	3	5
Application complexity	1/2	5	1/3	1	1/4
Organizational environment	2	3	1/5	4	1

1	Equally important	6	Strongly plus more important
2	Weakly more important	7	Very strongly more important
3	Moderately more important	8	Very, very strongly more important
4	Moderately plus more important	9	Extremely more important
5	Strongly more important		

Figure 20. DeltaAHP Questionnaire.

The best way to apply ΔAHP within a GDSS environment is by relying on questionnaires using a graphic representation of Saaty's semantic scale. To exemplify, consider a group of 8 decision-makers trying to assess the risks facing a software development project. The goal is then to assess software performance risk by developing a consensus of weights to be assigned to different factors. Assuming that the ν factors (criteria) contributing to software performance risk were defined as 1) technological newness, 2) application size, 3) expertise, 4) application complexity, and 5) organizational environment; $\nu(\nu-1)/2 = 5(5-1)/2 = 5\times4/2 = 10$ (greater than $\nu-1=4$) pairwise comparisons are required as follows: 1-2, 1-3, 1-4, 1-5, 2-3, 2-4, 2-5, 3-4, 3-5, and 4-5. Each decision-maker would have to complete a questionnaire similar to the one shown in Figure 20.

Using these pairwise comparisons, a judgment matrix for each decision-maker can be obtained. The first pairwise comparison (1-2: technological newness versus application size) yields 3 (i.e., the first decision-maker believes that technological newness is moderately more important than application size). Thus, a 3 is entered in row 1, column 2. For the second pairwise comparison (1-3: technological newness versus expertise), the decision-maker believes that technological newness is strongly **less** important than expertise, so the reciprocal of 5 (1/5) is entered in row 1, column 3.

Notice that some relationships show perfect consistency. For example, consider $e_{45} = 1/4 = e_{41}/e_{51} = (1/2)/2 = 1/4 = e_{41}/(1/e_{15})$. If as shown in cell e_{41}, application complexity (A) is weakly **less** important than technological newness (B), while at the same time cell $e_{15}=1/e_{51}$ tells us that technological newness (B) is weakly **less** important than organizational environment (C), then, by logic, application complexity (A) should be moderately plus **less** important than organizational environment (C) if the decision-maker is being "consistent" (i.e., if the information portrayed in the judgment matrix has been obtained through consistent reasoning). Symbolically, if $A < B$ and $B < C$, then $A < C$. This questionnaire-matrix format provides the basis to build an user-friendly GDSS interface. After all judgment matrices (one from each decision-maker) have been gathered, a final (consensus) judgment matrix and its uncertainty matrix have to be calculated. Summary statistics can be used for such purpose. The consensus matrix (**AA**) is obtained by calculating the average of each pairwise comparison for all decision-makers. Let $e_{ij,k}$ be the pairwise comparison between criterion i and criterion j obtained from decision-maker k ($k=1,\ldots,\zeta$). Then $e_{ij} = \Sigma e_{ij,k}/\zeta$ if $i \neq j$, 1 otherwise, $\forall$ $i=1,\ldots,\nu$, $j=1,\ldots,\nu$, $k=1,\ldots,\zeta$. The error matrix (**ΔAA**) is calculated dividing the standard deviation of each pairwise comparison for all decision-makers by the average pairwise comparison. Let Δe_{ij} be the error of the consensus pairwise comparison. As such, $\Delta e_{ij} = 100\text{x}[\Sigma(e_{ij,k}-e_{ij})^2/\zeta]/e_{ij}$ if $i \neq j$, 0 otherwise, $\forall$ $i=1,\ldots,\nu$, $j=1,\ldots,\nu$, $k=1,\ldots,\zeta$. A good statistical measure of how much of an agreement for each pairwise comparison has been reached by the group of decision-makers is the kurtosis (fourth moment of any distribution function). Kurtosis is a measure of peakedness. Let k_{ij} be the kurtosis for the set of ζ individual pairwise comparisons between criterion i and j. Then $k_{ij} = (\Sigma(e_{ij,k}-e_{ij})^4/\zeta)/(\Sigma(e_{ij,k}-e_{ij})^2/\zeta)^2$ if $i \neq j$, 0 otherwise, $\forall$ $i=1,\ldots,n$, $j=1,\ldots,n$, $k=1,\ldots,\zeta$. A mesokurtic curve ($k_{ij} = 1$) is an intermediate distribution. A platykurtic curve ($k_{ij} > 1$) is a broad or flat distribution. A leptokurtic curve ($k_{ij} < 1$) is a slender distribution. If consensus has been reached, most judgments are close and some are even the same (i.e., the frequency or probability discrete distribution of the sample data is peaked). We consider the result to be a consensus if at least the data is mesokurtic ($k_{ij} \leq 1$).

5.6 Discussion

Delta AHP can be used to calculate the relative weights of time, cost, and performance for the ILP model from the previous chapter (w_t, w_c, and w_q) based on the inputs of decision makers. Throughout the iterative modeling, validation, verification and implementation process followed in this chapter, it has been shown that the Gauss-Jordan numerical method cannot be used to calculate the inverse of the judgment matrix. The meaning of perfect consistency, as well as the relationship between consistency and invertibility have also been illustrated. The two reasons why ensuring perfect consistency is not recommended are: 1) the consistency ratio ought to be used as an indicator of the thoughtfulness of the decision-making process, and 2) the judgment matrix (**AA**) is singular if perfect consistency is assured by following the steps shown in Appendix B. A procedure to calculate the inverse matrix based on the definition of an inverse was analytically explained and successfully implemented based on the later inference. Finally, the algorithm used in Delta AHP was proved to lead to the same results as the VAX-BASIC code developed by Zahir.

Chapter 6. Conclusions

As we have seen in our models, time, cost, and performance are interrelated. Time and cost share a tradeoff illustrated by equation (51). There is also a tradeoff between time/cost and performance illustrated by equations (77) and (78). Making a decision on time such as crashing critical activities would have an impact on cost and performance. Therefore, decisions should be considered in light of all three variables and not just one or two. Risk adds another dimension at the portfolio level and should be considered as well, but the risk profile must be developed through a consensus of decision makers (see chapter five).

In chapter two we discussed one equation that correlates the mean with the corresponding shape parameters of beta distributed duration times. Based on that equation we derived two equations to calculate the shape parameters from the range parameters and the mean of beta distributed duration times. The equations are valid and can be applied to calculate the probabilistic completion time of activities in series. It is our recommendation to the practitioner to avoid using the PERT textbook formula for the mean, because of the underlying simplistic assumptions. We also recommend to calculate the mean and variance at joint nodes by numerical integration instead of using the PERT equation (maximum of the expectations) to avoid optimistic planning.

In chapter three we discussed the time-cost tradeoff and proposed a system of equations to minimize overall cost by crashing critical activities and delaying non-critical activities. Performance was included by considering the Net Present Value (NPV) and the Internal Rate of Return (IRR) of the project to be our measure of performance.

In chapter four we discussed a portfolio selection model based on zero-one integer linear programming. In practice, the optimal portfolio is a tradeoff decision between the model and investor's risk preferences. The priority index can be used in practice to help decide which projects to include in the portfolio.

In chapter five we discussed the Delta Analytic Hierarchy Process. We demonstrated that the Gauss-Jordan numerical method cannot be used to calculate the inverse of the judgment matrix and that the definition of the inverse has to be used instead. We also provided guidelines on how to avoid the construction of perfectly consistent matrices. Delta AHP can be used to determine the relative weight of time, cost, and performance for the risk constraint of the zero-one ILP model developed in chapter four.

Each chapter in this thesis addresses a particular topic. Combined, all these topics constitute our methodology. The methodology's algorithm is as follows:

Step 1. Calculate the expected completion time and its deviation as well as the criticality index for each activity (chapter 2).

Step 2. Calculate the cost for the expected duration of each activity and for the minimum and maximum duration times. Then, estimate the deviation as the minimum difference between the expected cost and the costs for the minimum and maximum duration times. Use that to calculate the optimal schedule and to assign slacks (chapter 3).

Step 3. Calculate the performance (Internal Rate of Return) for the expected, minimum, and maximum times and costs and estimate the deviation as the minimum difference between the expected and the minimum and maximum figures (chapter 3).

The algorithm above applies to individual projects. To consider a set of alternative projects (portfolio) requires applying the following algorithm:

Step 1. Apply the Delta Analytic Hierarchy Process (ΔAHP) to determine the relative importance of the time, cost, and performance dimensions (chapter 5).

Step 2. Calculate the priority index for each project to help decision makers interact with the solution (chapter 4).

Step 3. Apply the methodology to each and every project available and then solve the zero-one integer linear programming model to obtain the "optimal" portfolio (chapter 4).

Although our methodology requires more effort than PERT, it provides estimates for not only time, but also cost and performance (where performance is defined as the projects Internal Rate of Return). Furthermore, it considers the tradeoff between time and cost and allows for the possibility of calculating the "optimal" schedule.

To recapitulate, our methodology attempts to quantify the time-cost-performance tradeoff that exists in all projects. Our mathematical model addresses the project management questions for it considers risk plus all project management dimensions: time, cost, and performance. However, it is no substitute for common sense and best practices. The novelty we bring is the equation to accurately calculate beta distributed duration times from the minimum, mean, and maximum duration times. We also provide a holistic point of view that considers all project management dimensions at once: time, cost and performance.

Further research should consider risk in more detail, in particular, the existence of time-risk, cost-risk, and performance-risk, and their interactions, as well as overall risk. It may be of interest to consider the display of information and the kinds of charts that can be used to portray results from the model and to allow decision makers to interact with the solution and change the result suggested by the model if considered necessary. Integrating the mathematical model proposed in this thesis with a specific interface design aimed at creating a Group Decision Support System (GDSS) is still left for further work. Further releases of Delta AHP will include the standard AHP algorithm of successive squaring as well as the questionnaire-based interface to speed up information gathering. Also, left for further research is sensitivity analysis using realistic project data to test the outcomes and use it in practical situations. Finally, it may prove useful to revise all the concepts in chapters two and three and present a synthesis from the point of view of capital investment decision analysis.

References

[1] Adams, Carl, "Risk aversion", *Project Finance*, pp. 2-5 (June 2001).

[2] Archer, N.P., and F. Ghasemzadeh, "An integrated framework for project portfolio selection", *International Journal of Project Management*, Vol. 17, No. 4, pp. 207-216 (1999).

[3] Archer, Norman P., and Fereidoun Ghasemzadeh, "A decision support system for project portfolio selection", *International Journal of Technology Management*, Vol. 16, Nos. 1/2/3, pp. 105-114 (1998).

[4] Arrington, C. Edward, "An application of the analytical hierarchy process to model expert judgments on analytical review procedures", *Journal of Accounting Research*, Vol. 22, Issue 1, pp. 298-312 (1984).

[5] Auguston, Karen A., "Activity-based costing adds up to big benefits", *Modern Materials Handling*, Vol. 48, Issue 7, pp. 57-59 (1993).

[6] Babu, A.J.G., and Nalina Surech, "Project management with time, cost, and quality considerations", *European Journal of Operational Research*, Vol. 88, Issue 2, pp. 320-327 (1996).

[7] Baram, George E., "Concurrent delays –what are they and how to deal with them?", *AACE International Transactions*, pp. R7A-R14A (2000).

[8] Basak, Indrani, "Probabilistic judgments specified partially in the analytic hierarchy process", *European Journal of Operational Research*, Vol. 108, Issue 1, pp. 153-164 (1998).

[9] Bertin, Jacques. *Graphics and Graphic Information Processing*. Berlin: Walter de Gruyter (1981).

[10] Bertin, Jacques. *Semiology of Graphics: Diagrams, Networks, Maps.* London: The University of Wisconsin Press (1983).

[11] Brealey, Richard A., and Stewart C. Myers., *Principles of Corporate Finance*, New York: McGraw Hill (1991).

[12] Burden, R.L., and J.D. Faires, *Numerical Analysis*, Boston: PWS-Kent Pub. Co (1993).

[13] Bussey, Lynn E., *The Economic Analysis of Industrial Projects*, Englewood Cliffs, New Jersey: Prentice Hall (1978).

[14] Canada, J.R., W.G. Sullivan, and J.A. White, *Capital Investment Analysis for Engineering and Management*, Upper Saddle River, NJ: Prentice Hall (1996).

[15] Chachra, V., P. Ghare, and J. Moore, *Applications of Graph Theory Algorithms*, New York: North Holland (1979).

[16] Chapman, Chris, and Stephen Ward, *Project Risk Management: Processes, Techniques and Insights*, John Wiley & Sons (1997).

[17] Clark, Ralph H., "Benefit/Cost Analysis of Trenchless Alternatives", *Transactions of the American Association of Cost Engineers*, Vol. 2, pp. Q.1.1-Q.1.8 (1992).

[18] Cleveland, William S. *Visualizing Data.* Murray Hill, NJ: AT&T Bell Laboratories (1993).

[19] Cleveland, William S., and Robert McGill. "Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods", in *Journal of the American Statistical Association*, Vol. 79, No. 387 (September 1984), pp. 531-554.

[20] Cooper, R.G., S.J. Edgett, and E.J. Kleinschmidt, *Portfolio Management for New Products*. Hamilton, Ontario: McMaster University (1997).

[21] Dawson, R.J., "Practical proposals for managing uncertainty and risk in project planning", *International Journal of Project Management*, Vol. 16, Issue 5, pp. 299-310 (1998).

[22] Denker, Stephen, "Planning concurrency and managing iteration in projects", *Project Management Journal*, Vol. 32, Issue 3, pp. 31-38 (2001).

[23] Devroye, Luc. P., "Inequalities for the completion times of stochastic PERT networks", *Mathematics of Operations Research*, Vol. 4, No. 4, pp. 441-447 (1979).

[24] Dey, Prasanta K., "Project time risk analysis through simulation", *Cost Engineering*, Vol. 43, Issue 7, pp. 24-32 (2001).

[25] Dodin, Bajis M., and Salah E. Elmaghraby, "Approximating the criticality indices of the activities in PERT networks", *Management Science*, Vol. 31, No. 2, pp. 207-223 (1985).

[26] Dodin, Bajis, "Bounding the project completion time distribution in PERT networks", *Operations Research*, Vol. 33, Issue 4, pp. 862-881 (1985).

[27] Dodin, Bajis, "Determining the k most critical paths in PERT networks", *Operations Research*, Vol. 32, No. 4, pp. 859-877 (1984).

[28] Donaldson, William A., "The estimation of the mean and variance of a 'PERT' activity time", *Operations Research*, Vol. 13, pp. 382-385 (1965).

[29] Edwards, R.E., *Functional Analysis: Theory and Applications*, Holt, Rinehart & Winston: New York (1965).

[30] Emshoff, James R., and Thomas L. Saaty, "Applications of the analytic hierarchy process to long range planning processes", *European Journal of Operational Research*, Vol. 10, Issue 2, pp. 131-143 (1982).

[31] Evans, T., and E. Minieka, *Optimization Algorithms for Networks and Graphs*, New York: Dekker (1992).

[32] Farnum, N.R., and L.W. Stanton, "Some results concerning the estimation of beta distribution parameters in PERT", *Journal of the Operational Research Society*, Vol. 38, pp. 287-290 (1987).

[33] Farrell, L.M., "The impact of illiquidity and uncertainty of the multiperiod project investment decision process", *Project Management Journal*, Vol. 27, Issue 3, pp. 35-45 (1996).

[34] Fisher, Donald L., Donna Saisi, and William M. Goldstein, "Stochastic PERT networks: OP diagrams, critical paths and the project completion time", *Computers & Operations Research*, Vol. 12, Issue 5, pp. 471-482 (1985).

[35] Foldes, Stephan, and Francois Soumis, "PERT and crashing revisited: mathematical generalizations", *European Journal of Operational Research*, Vol. 64, Issue 2, pp. 286-294 (1993).

[36] Ghasemzadeh, Fereidoun, Norm Archer, and Paul Iyogun, "A zero-one model for project portfolio selection and scheduling", *Journal of the Operational Research Society*, Vol. 50, pp. 745-755 (1999).

[37] Golenko-Ginzburg, D., "On the distribution of activity time in PERT", *Journal of the Operational Research Society*, Vol. 39, pp. 767-771 (1988).

[38] Grubbs, Frank E., "Attempts to validate certain PERT statistics or 'picking on PERT'", *Operations Research*, Vol. 10, pp. 912-915 (1962).

[39] Halton, John H., "A retrospective and prospective survey of the Monte Carlo method", *SIAM Review*, Vol. 12, No. 1, pp. 1-63 (1970).

[40] Hammersley, J.M., and D.C. Handscomb, *Monte Carlo Methods*, Methuen: London (1964).

[41] Hastings, N.A.J., and J.B. Peacock, *Statistical Distributions*, New York: John Wiley & Sons (1975).

[42] Hauser, David, and Pandu Tadikamalla, "The analytic hierarchy process in an uncertain environment: a simulation approach", *European Journal of the Operational Research*, Vol. 91, Issue 1, pp. 27-37 (1996).

[43] Hoedemaker, Geert M., "Limits to currency", *Decision Sciences*, Vol. 30, Issue 1, pp. 1-18 (1999).

[44] Icmeli, Oya, "The resource constrained time/cost tradeoff project scheduling problem with discounted cash flows", *Journal of Operations Management*, Vol. 14, Issue 3, pp. 255-275 (1996).

[45] Johnson, George A., and Corey D. Schou, "Expediting projects in PERT with stochastic time estimates", *Project Management Journal*, Vol. 21, Issue 2, pp. 29-34 (1990).

[46] Kamenetzky, Ricardo D., "The relationship between the analytic hierarchy process and the additive value function", *Decision Sciences*, Vol. 13, Issue 4, pp. 702-713 (1982).

[47] Keefer, Donald L., and William A. Verdini, "Better estimation of PERT activity time parameters", *Management Science*, Vol. 39, pp. 1086-1091 (1993).

[48] Klein, Jonathan H., "Modeling Risk Trade-Off", *Journal of the Operational Research Society*, Operational Research Society, Vol. 44, No. 5, pp. 445-460 (1993).

[49] Kleindorfer, G.B., "Bounding Distributions for a Stochastic Acyclic Network", *Operations Research*, Vol. 19, 1589-1601 (1971).

[50] Kraft, R., and C.J. Wensrich, "Monte Carlo methods – a bibliography covering the period 1947 to June 1961", *Rep. UCRL 6581*, Lawrence Radiation Laboratory, University of California, Berkeley (1961).

[51] Lawler, E., *Combinatorial Optimization: Networks and Matroids*, Chicago: Holt, Rinehart & Winston (1976).

[52] Lefley, Frank, "Approaches to risk and uncertainty in the appraisal of new technology capital projects", *International Journal of Production Economics*, Vol. 53, Issue 1, pp. 21-43 (1997).

[53] Lohse, Gerald L., et al. "A Classification of Visual Representations", in *Communications of the ACM*, Vol. 37, No. 12 (December 1994), pp. 36-49.

[54] MacCrimmon, Kenneth R., and Charles A. Ryavec, "An analytical study of the PERT assumptions", *Operations Research*, Vol. 12, pp. 16-37 (1964).

[55] Malcolm, D.G., J.H. Roseboom, C.E. Clark, and W. Fazar, "Application of a Technique for Research and Development Program Evaluation", *Operations Research*, Vol. 7, 646-669 (1959).

[56] Martin, J., "Distribution of the Time Through a Directed, Acyclic Network", *Operations Research*, Vol. 13, 46-66 (1965).

[57] McManus, John, "Risk in software projects", *Management Services*, Vol. 45, Issue 10, pp. 6-10 (2001).

[58] Meredith, Jack R., and Samuel J. Mantel, Jr., *Project Management: A Managerial Approach*, New York: John Wiley & Sons (1995).

[59] Metropolis, M., and S.M. Ulam, "The Monte Carlo method", *Journal of the American Statistical Association*, Vol. 44, pp. 335-341 (1949).

[60] Ockham, William, *Philosophical Writings: A Selection*, Indianapolis: Hackett Pub. Co. (1990).

[61] Parzen, Emanuel, *Modern Probability Theory and its Applications*, New York: John Wiley and Sons (1960).

[62] Phillips, D., and A. Diaz, *Fundamentals of Network Analysis*, Englewood Cliffs, N.J.: Prentice Hall (1981).

[63] Platge, Adri, Harald Seidel, and Sipke Wadman, "Project and Portfolio Planning Cycle", *International Journal of Project Management*, Vol. 12, Issue 2 (1994), pp. 100-106.

[64] Powell, Antony, "Strategies for lifecycle concurrency and iteration – A system dynamics approach", *The Journal of Systems and Software*, Vol. 46, Issue 2, pp. 151-161 (1999).

[65] Preece, Jenny, et al. *Human-Computer Interaction*. Addison-Wesley (1996).

[66] Ries, Al, and Jack Trout, "Focus in a Fuzzy World", *Executive Excellence*, Vol. 17, Issue 14, pp. 3-4 (2000).

[67] Rivenbank, William C., "Benchmarking and cost accounting: the North Carolina approach", *Journal of Public Budgeting, Accounting & Financial Management*, Vol. 12, Issue 1, pp. 125-147 (2000).

[68] Rose, Aaron, "The role of probability in scheduling", *American Association of Cost Engineers – Transactions of the American Association of Cost Engineers*, pp. H.2.1-H.2.8 (1981).

[69] Rosenbloom, E.S., "Theory and Methodology: A probabilistic interpretation of the final rankings in AHP", *European Journal of Operational Research*, Vol. 96, pp. 371-378 (1996).

[70] Saaty, T.L., "A Scaling Method for Priorities in Hierarchical Structures", *Journal of Mathematical Psychology*. Vol. 15 (1977), pp. 234-281.

[71] Saaty, Thomas L., "Highlights and critical points in the theory and application of the analytic hierarchy process", *European Journal of Operational Research*, Vol. 74, Issue 3, pp. 426-447 (1994).

[72] Saaty, Thomas L., *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*, Pittsburgh: RWS Publications (1990).

[73] Sasieni, M.W., "A note on PERT times", *Management Science*, Vol. 32, No. 12, pp. 1652-1653 (1986).

[74] Schimmoller, Brian K., "The shifting sands of risk management", *Power Engineering*, Vol. 105, Issue 9, p. 3 (2001).

[75] Schoner, Bertran, William C. Wedley, and Eng Ung Choo, "A unified approach to AHP with linking pins", *European Journal of Operational Research*, Vol. 64, Issue 3, pp. 384-372 (1993).

[76] Shreider, Yu A., *Method of Statistical Testing (Monte Carlo Method)*, Elsevier: Amsterdam (1964).

[77] Simon, H. *The New Science of Management Decisions*, Englewood Cliffs, NJ: Prentice Hall (1977).

[78] Smolin, Rocky, "A critical path method for project control", *Interface Age, Computing for Business*, Vol. 6, Issue 8, pp. 106-110 (1981).

[79] Sunde, Leif, and Steen Lichtenberg, "Net-present value cost/time tradeoff", *International Journal of Project Management*, Vol. 13, Issue 1, pp. 45-49 (1995).

[80] Tattersall, Jim, "Understanding overheads – their management and control", *Management Decision*, Vol. 28, Issue 4, pp. 4-8 (1990).

[81] Tavares, L. Valadares, "On the optimal management of project risk", *European Journal of Operational Research*, Vol. 107, Issue 2, pp. 451-469 (1998).

[82] Terwiesch, Christian, "Measuring the effectiveness of overlapping development activities", *Management Science*, Vol. 45, Issue 4, pp. 455-465 (1999).

[83] Touran, A., and E.P. Wiser, "Monte Carlo technique with correlated variables", *Journal of Construction Engineering and Management*, Vol. 118, Issue 2, pp. 258-272 (1992).

[84] Turban, Efraim, and Jay E. Aronson, *Decision Support Systems and Intelligent Systems*, (1995).

[85] Van Slyke, Richard, "Monte Carlo methods and the PERT problem", *Operations Research*, Vol. 11, Issue 5, pp. 839-860 (1963).

[86] Walker, Larry E., "Advantages of risk-based project management", *Occupational Health & Safety*, Vol. 70, Issue 9, pp. 161-164.

[87] Walter, J.K., "The Spectrum of Mathematical Models", *Perspectives in Computing*, Vol. 3, pp. 4-13 (May 1983).

[88] Wiest, J., and F. Levy, *A Management Guide to PERT/CPM*, Englewood Cliffs, N.J.: Prentice Hall (1997).

[89] Winston, Wayne L., *Operations Research: Applications and Algorithms*, Belmont, California: Duxbury Press (1994).

[90] Wolf, Gary L., "PERT/CPM: A tool for managing projects", *Industrial Management*, Vol. 27, Issue 1, pp. 22-25 (1985).

[91] Zahir, M.S., "Incorporating the uncertainty of decision judgments in the analytic hierarchy process", *European Journal of Operational Research*, Vol. 53, pp. 206-216 (1991).

[92] Zaremba, S.K., "The mathematical basis of Monte Carlo and quasi-Monte Carlo methods", *Review*, Vol. 10, pp. 303-314 (1968).

Appendix A. Gauss Jordan Inversion Process in AHP

Let **A** be the judgment matrix and **I** be the identity matrix. By definition, **AI** = **A**. Let **G** be the augmented matrix **A|I**, which is formed by merging the two nxn matrices into a single nx2n matrix, having g_{ij} as its constituent elements, ∀ i=1,…,n, j=1,…,2n. Using row operations, it is usually possible to transform the left part of the augmented matrix **G** into the identity matrix. The matrix that remains at the right is $\mathbf{A}^{-1}$. In linear algebra, this would be equivalent to say that ax1=a, where a is any real number. To transform a into 1 we would have to multiply a by its reciprocal 1/a (inverse in matrix notation). This modifies our original equation from ax1=a into 1x(1/a)=1/a, where 1/a is the "inverse" of a, since ax1/a = 1 (in matrix notation, $\mathbf{AA}^{-1} = \mathbf{I}$). To apply this concept, we have to modify the left side of **G** column by column. The first step is to obtain a 1 in row/column 1/1 by multiplying the entire row by $1/g_{11}$. Then, a sequence of n-1 row sums is carried out multiplying row 1 by $-g_{i1}$ and then adding row 1 to row i, ∀ i=2,…,n. The next step is to obtain a 1 in row/column i/j, where i=j. To do so, row i is multiplied by $1/g_{ii}$. The same procedure is repeated for all the remaining n-1 left columns of **G**. If successful (no divisions by zero), this results in a modified augmented matrix **G**, which contains the identity matrix **I** at the left side and the inverse matrix of **A** at the right side ($\mathbf{I}|\mathbf{A}^{-1}$).

The first step (multiplying row 1 by $1/c_{11}$) is not required, since we already have a 1 there. The second step involves adding to row 2 row 1 multiplied by $-1/c_{12}$. This guarantees a zero in row/column 2/1 (which is what we need), but leaves also a zero in row/column 2/2, because row/column 2/2 equals 1 (which is what it already has), plus row/column 1/2 (c_{12}) multiplied by ($-1/c_{12}$), which equals to $1-c_{12}/c_{12} = 1-1 = 0$. Now, it is impossible to obtain a 1 in row/column 2/2, because doing so would result in a division by zero (see Figure 21).

Such limitation of the Gauss-Jordan method is relevant for system design purposes. For example, if the attribute entered in row/column 1/2 is 3, its reciprocal in row column 2/1 would be approximately equal to 0.33. Nevertheless, it was found that if a number large enough of this kind of cases occurred, the inverse matrix could still be obtained by the Gauss-Jordan numerical method when the main diagonal of the left-side augmented matrix results in numbers approximately equal to zero, but not zero (1/3 = 0.33 is not zero, but 0.0033333333 is almost zero). This may not guarantee a good approximation to the inverse matrix of **A** due to mantissa errors. Further tests were carried out and the input format shown in Figure 19 was modified to allow users to enter zeroes when the exact reciprocal in row/column i/j ought to be obtained from row/column j/i ($c_{ij}=1/c_{ji}$). The result was the inability of the system to calculate the inverse matrix. As a consequence, the entire ΔAHP methodology and the underlying math were reviewed for further considerations in order to decide how to calculate the inverse matrix (refer back to Figure 15).

$$\mathbf{G = A \mid I}$$

$$\left[\begin{array}{ccccc|ccccc}
1 & c_{12} & c_{13} & \cdots & c_{1n} & 1 & 0 & 0 & \cdots & 0 \\
\frac{1}{c_{12}} & 1 & c_{23} & \cdots & c_{2n} & 0 & 1 & 0 & \cdots & 0 \\
\frac{1}{c_{13}} & \frac{1}{c_{23}} & 1 & \cdots & c_{3n} & 0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\frac{1}{c_{1n}} & \frac{1}{c_{2n}} & \frac{1}{c_{nn}} & \cdots & 1 & 0 & 0 & 0 & \cdots & 1
\end{array}\right]$$

Figure 21. Inverse of Judgment Matrices with the Gauss-Jordan Numerical Method.

Appendix B. Ensuring Perfect Consistency

Building perfectly consistent matrices is precisely what decision-makers should avoid. The purpose of this algorithm is to show what should be avoided. As a rule of thumb it is safe to proceed when considerably more than n-1 pairwise comparisons have been requested.

1) Request n-1 pairwise comparisons from decision-makers and place them in the judgment matrix accordingly.
2) At least two of these pairwise comparisons have to be in the same row or column.
3) If the i^{th},j^{th} pairwise comparison (attribute a_{ij}) is above the main diagonal, a column should be canceled. If a_{ij} is below the main diagonal, a row is canceled. New pairwise comparisons cannot be entered in canceled spots.
4) Reciprocals should be automatically calculated, but no row or column cancellation for the reciprocal has to be made.
5) For all the remaining n(n-1) - 2(n-1) = (n-1)(n-2) attributes, it will always be possible to calculate them by using the perfect consistency equation $c_{ij} = c_{ik}/c_{jk}$. For each c_{ij} attribute calculated in such way, its reciprocal can be automatically calculated as well. The perfect consistency equation would have to be used (n-1)(n-2)/2 times.

Appendix C. Visualizing Project Management Dimensions

Introduction

Visualization includes the study of both image synthesis and image understanding, spanning many academic disciplines, scientific fields, and multiple domains of inquiry. Lohse et al. ([53]) argue that the need for classification schemes is based on the fact that classification is at the heart of every scientific field. Classifications "structure domains of systematic inquiry and provide concepts for developing theories to identify anomalies and to predict future research needs". Graphs and images can be used to visualize classifications and different dimensions. Graphs and images can be characterized as either functional (focus on the intended use and purpose of the graphic material) or structural (focus on the form of the image rather than its content). Graphs encode quantitative information using position and magnitude of geometric objects. Numerical data in one, two, or three dimensions are plotted on a Cartesian coordinate or polar coordinate system. Common graph types include scatterplot, categorical, line, stacked bar, bar, pie, box, fan, response surface, histogram, star, polar coordinate, and Chernoff face graphs. Preece et. al ([65]) identify seven techniques to represent numeric data: scatterplots, line graphs or curves, area, band, strata or surface charts, bar graphs, column charts or histograms, pie charts, simulated meters, and star, circular or pattern charts. In this work, we use a modified version of scatterplots in a Cartesian coordinate system with position, length and area judgments for information coding.

Projects are typically specified using three dimensions: time, cost, and performance. We add another dimension, risk, to consider the probabilistic nature of the portfolio selection process. In the following sections, the reasoning process followed and the interface features of two alternative displays are discussed based on the literature reviewed (theory). The above leads into the conceptualization of two interfaces to choose from (design). A simple experiment is devised (testing) to determine which interface performs better in each circumstance (discussion).

Theory

Bertin ([10]) defines understanding as "simplifying, reducing a vast amount of «data» to the small number of categories of «information» that we are capable of taking into account in dealing with a given problem". Preece et al. ([65]) discuss what is known in the Human-Computer Interaction (HCI) literature as the 7±2 magic number, related to short term memory, which shows that humans are able to recall between 5 and 9 numbers or figures at the same time. That is one of the reasons why a good HCI display is critical, since it allows users to consider several numbers all at once if the display presents information in a meaningful way. Although this concept regarding understanding seems to be most accurate, the human brain is actually much more capable than Bertin seems to imply. The mind is able to make abstractions, synthesize various elements from reality, and put them together using not only short-term, but also long-term memory. A suitable design is a polysemic graphic system, in which the meaning of the individual signs follows and is deduced from consideration of the collection of signs. For our purposes, perception deals with the ability of any given individual (or group of) expert(s) to find relationships between the images and the real world, in an attempt to reach the best project portfolio. The information displayed on the screen or printed on a sheet of paper is the result of summarizing in a plot, based on mathematical models, the combination of all the available data from historical records (retrieved from a database) and the input obtained from experts during each session.

For any given project, we have four dimensions: time, cost, performance, and risk. Let t_k, c_k, p_k, and r_k be the time, cost, performance and risk of project k for all k=1…s, where s is the total number of projects (portfolio size). Also, let the zero-one decision variable x_k indicate whether or not project k is selected; if x_k=0 project k is not selected, if x_k=1 project k is selected. In vector notation, $\mathbf{T} = \{t_1, t_2, \ldots, t_s\}$, $\mathbf{C} = \{c_1, c_2, \ldots, c_s\}$, $\mathbf{P} = \{p_1, p_2, \ldots, p_s\}$, $\mathbf{R} = \{r_1, r_2, \ldots, r_s\}$, and $\mathbf{X} = \{x_1, x_2, \ldots, x_s\}$. Vectors **T**, **C**, **P**, and **R** are row vectors, whereas vector **X** is a column vector. These estimates are the result of applying mathematical models to the raw (input) data obtained from historical records and/or decision-makers' expertise. For the purposes of our empirical study, these figures are given. The objective is to maximize performance subject to cost and risk constraints. The cost constraint is not to exceed the budget (B); the risk constraint is not to exceed, on average, investor's risk preferences (K). Equations (99) to (101) portray the simplified zero-one integer programming model used in our study to find the optimal solution (see Ghasemzadeh, Archer, and Iyogun [36] for a detailed discussion).

Maximize $$\mathbf{PX} \tag{99}$$

Subject to:

$$\mathbf{CX} \leq \mathrm{B} \tag{100}$$

$$(\mathbf{R} - \mathrm{K})\mathbf{X} \leq 0 \tag{101}$$

We have to meaningfully portray 4 dimensions of data. In table format we have s rows and four columns; each row represents a project and each column a project dimension. The problem we have is what Bertin ([9]) calls the impassable barrier: with up to three columns, a data table can be constructed directly as a single image, producing a scatterplot or correlation diagram, in which the objects are in the third (vertical dimension) typically denoted as the z axis. But we have four (not three) dimensions to picture. Is there anything we can do to avoid sacrificing the overall relationships of the entire set?

Considering what can be represented in a flat sheet of paper, a graphic system can include eight variables besides the two to three axes of the plane or space: a) size, b) value, c) texture, d) color, e) orientation, and f) shape. Cleveland and McGill ([19]) ordered from most to least accurate the ten elementary perceptual tasks:

1) Position along a common scale.
2) Positions along nonaligned scales.
3) Length, direction, angle.
4) Area.
5) Volume, curvature.
6) Shading or color saturation.

Our main problem is how to portray 4 dimensions. Cleveland ([18]) explores the use of multiple scatterplots in a multipanel display of 4 rows and 4 columns for hypervariate data. Each pair of variables is graphed on a scatterplot within each panel; the left column is column one and the bottom row is row one. The graphs are arranged in a shared-scales matrix: along each row or column, one variable is plotted against all others, which allows to visually link features of one scatterplot with another. The data is pictured by relying on position along a common scale: small circles correlate each pair of variables (see Figure 22).

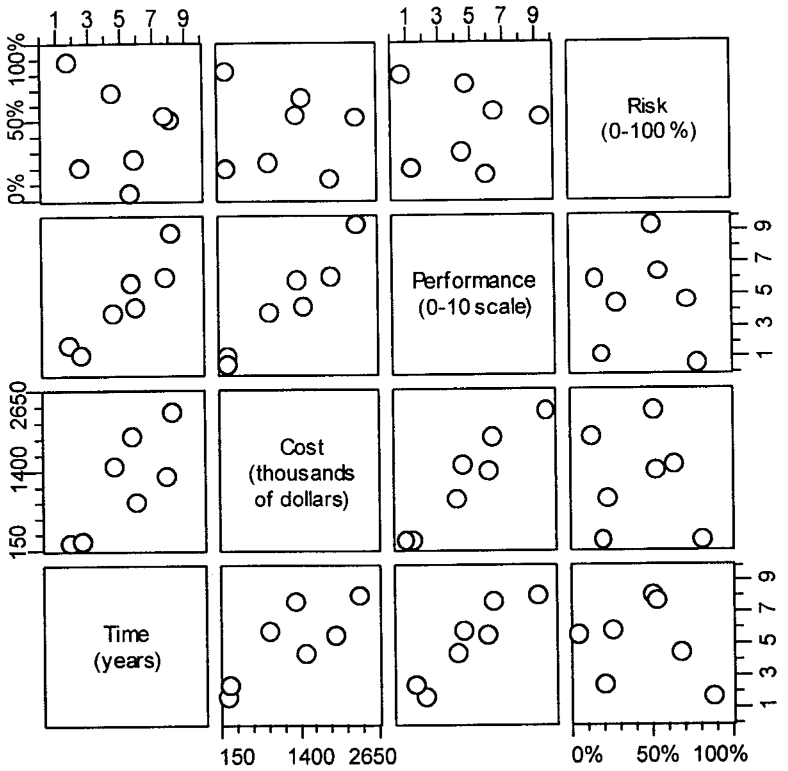


Figure 22. Multipanel Scatterplots.

The display in Figure 22 shows too much information at once. Bertin ([9]) illustrates the use of bubbles to picture position along the row/column common scale in multiple scatterplots. The diameter (as opposed to the area) of the bubbles pictures another dimension because length is more accurate than area. The scatterplot shown in Figure 22 can be modified by portraying the first two dimensions (time and cost) as well as risk along the common scales, depicting performance as the diameter of each bubble. Intuitively, a good portfolio minimizes time and cost while maximizing performance for a given risk profile. Risk should not be portrayed as area because the risk profile is usually expressed as a risk range, and position along a common scale allows a more accurate perception than length. Since time and cost share the same objective (minimization), it seems reasonable to use the same elementary perceptual task (position along a common scale) in both cases. The remaining variable is performance, which is portrayed using the diameter of the bubble as shown in Figure 23.

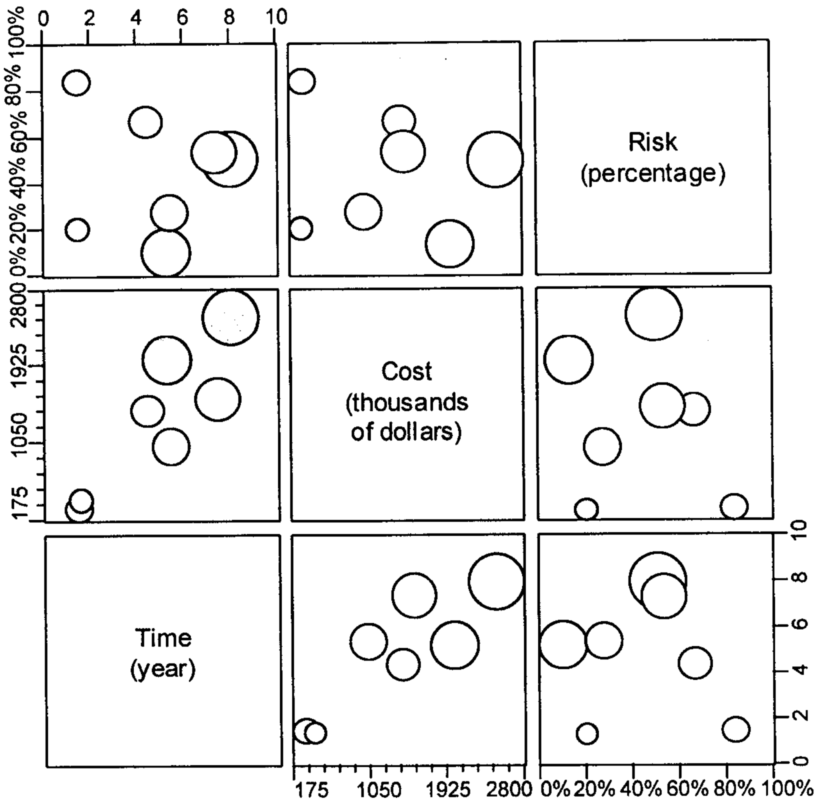


Figure 23. Multipanel Bubble Scatterplots.

Design

We are trying to measure interface efficiency, where efficiency is defined as the percentage of correct portfolio selections from the user using a given display such as the one from Figure 23 when compared to the optimal solution according to the model from equations (99) to (101). Although Figure 23 meaningfully displays all four dimensions at once, it would be difficult to make sense of the information for larger portfolios, because the bubbles would interfere with each other or if the display were scaled down, the resulting bubbles would be too small to be clearly visible. Even if each display is clear, excessive information may clutter the computer display and confuse the user. The solution is to choose which dimension to portray. But which of the four dimensions should we choose for each axis? Looking back at the decision support model from equations (99) to (101) we realize that only cost and risk are constraints. Thus, it makes sense to assign risk and cost to the horizontal and vertical axes, respectively, leaving performance (the objective function) to be depicted by the diameter of the bubbles, ignoring time as being irrelevant in this particular case. Users will be advised to choose projects as close to the horizontal axis (i.e. as low in cost) and as large in diameter as possible while at the same time keeping the average risk below the maximum allowed.

Another way to improve the display is by substituting framed rectangles for bubbles. According to Cleveland and McGill ([19]), position along non-aligned scales is more accurate than length. The height of a rectangle can be used to portray performance instead of the diameter of circles. To portray information more accurately we frame the rectangle within another rectangle in such way that a sense of percentage can be obtained by comparing the relative distances between the inner and the outer rectangle. Let project j be the project of the maximum performance; then the performance of project i is given as a percentage of that as shown in Figure 24.

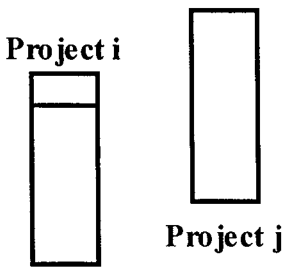


Figure 24. Framed Rectangles.

Consequently, we have two alternative displays. The first display (see Figure 25) is the bubble chart display and it portrays performance of each portfolio as the diameter of each bubble. The second display (see Figure 26) is the frames chart, portraying project performance as the height of the rectangle, which provides an idea of percentage because the inner rectangle can always be compared to the outer rectangle.

Our hypothesis is that the second display (frame chart) should portray data better for representing performance using positions along nonaligned scales (rectangle within rectangle idea) as opposed to the use of length of the first display (bubble chart). Furthermore, such interface improvement should translate into a better ability of the user to interact with the display and to select portfolios closer to the optimal solution.

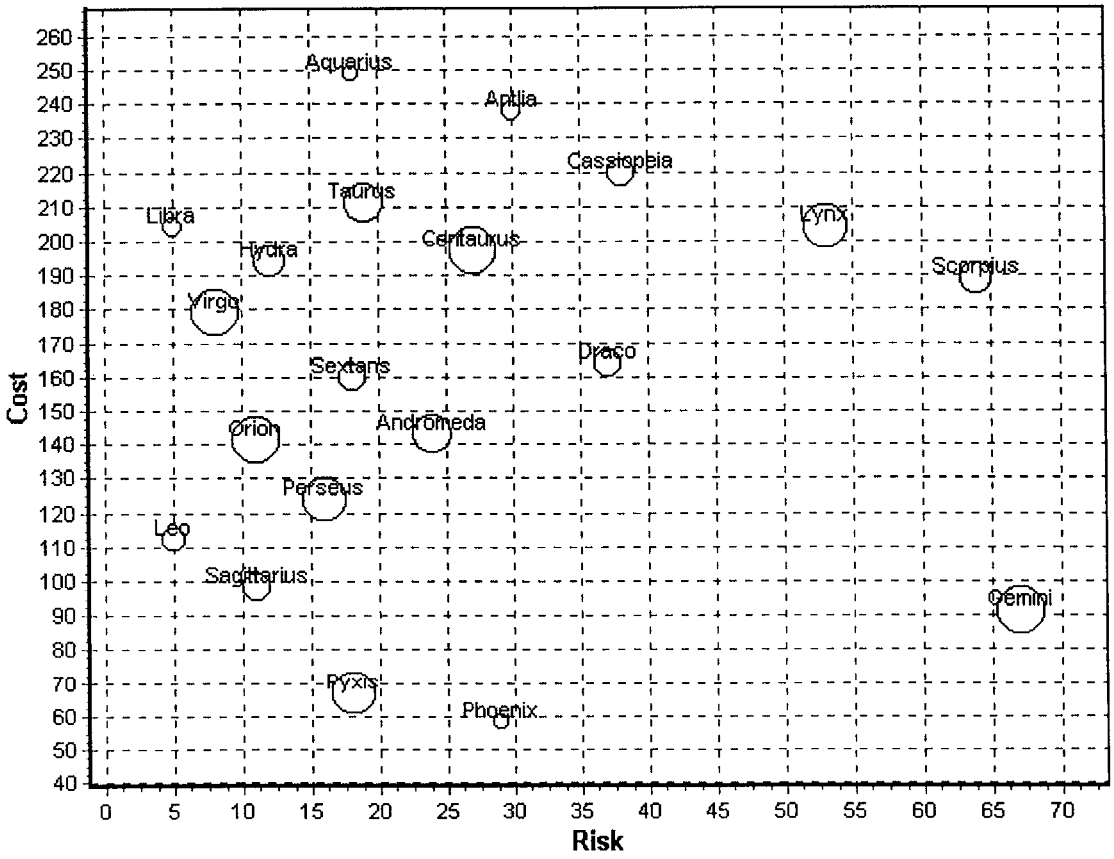


Figure 25. Bubble Chart.

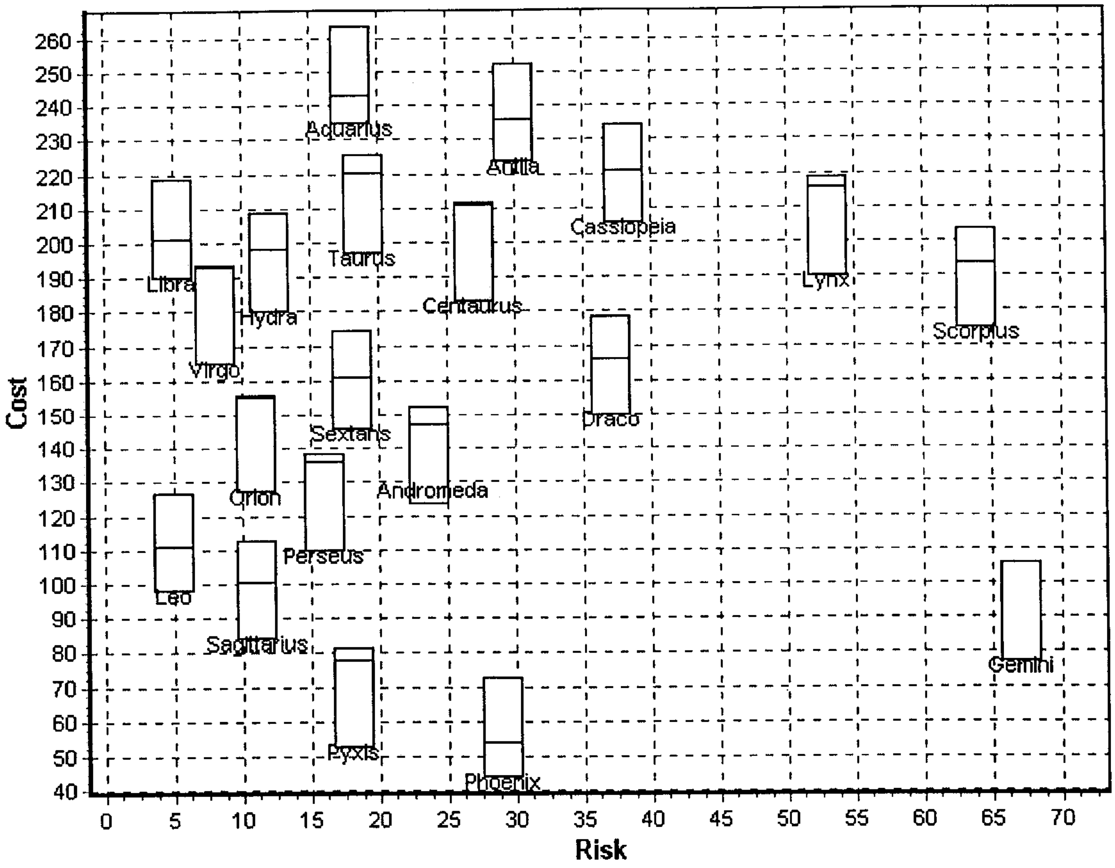


Figure 26. Frame Chart.

Testing

We devised a test to decide which interface display (bubbles chart in Figure 25 versus frames chart in Figure 26) performs better. Performing better in the context of our discussion means that, on average, the display leads users to solutions closer to the optimal. Thus, to decide which display is better, we need to know the error for a given portfolio and the average error for a given display.

The error (E) for a given portfolio is the sum of the squared differences between the solution vector and the data vector. Solution (**S**={S_k for all k=1...n}) and data (**D**={D_k for all k=1...n}) vectors are binary vectors of size n, where n indicates portfolio size. S_k=1 in the solution vector means that project k is part of the optimal solution; conversely, S_k=0 indicates that project k should be kept out of the solution vector. D_k=1 means that the user selected project k as part of the portfolio; D_k=0 indicates that project k was not included in the portfolio. The error (E) as shown in equation (102), indicates the total number of misallocations: either the user selected a project that should have not been selected, or the user forgot to select a project that had to be selected.

$$E = \sum_{k=1}^{n} (S_k - D_k)^2 \tag{102}$$

Values for E occur for the data of each and every subject in the experiment. Let m be the total number of subjects and E_x denote the error on experiment x. Then, the mean (average) error can be calculated as the average of errors as shown in equation (103).

$$\overline{E} = \frac{\sum_{x=1}^{m} E_x}{m} \tag{103}$$

The average error can be used to compare the two displays because the best display is expected to have a lower error when compared to the alternative display.

Discussion

A total of 10 subjects participated in the pilot study. The pilot study involved interacting with both displays (one at a time) and deciding, based on the visual information displayed, which projects to select. All subjects were advised of the best strategy: go for the projects with the highest performance first. As it turns out, it is more difficult to identify which projects have the largest performance using the bubble chart. In fact, 7 out of 10 users said they preferred the frame chart compared to the bubble chart. On the other hand, the mean error for the bubble chart was 1.8 whereas the mean error for the frame chart was 2.1. The latter seems to indicate an advantage of the bubble chart compared to the frame chart; however, such advantage is not large enough to discard our hypothesis. The number of users who found the optimal solution is larger in the case of the frame chart (4 users found the optimal solution using the frame chart, whereas only 3 users found the optimal solution using the bubble chart).

Only 2 users complained they did not have enough time to complete the experiment. On a scale from 1 to 7 (where 1 is strongly disagrees and 7 is strongly agrees), the group rated the bubble chart as 5 and the frames chart as 5.44 in usefulness. In summary, it seems the frame chart is the favorite despite the fact that it did not perform better than the bubble chart. The small advantage in the mean error of the bubble chart is not sufficiently large to discard our hypothesis, but the evidence in favor of the frame chart is not solid enough to declare a winner.