

Software Packages for Multi-Criteria Resource Allocation

João Carlos Lourenço, Carlos A. Bana e Costa, Alec Morton

Abstract—In this paper four commercial software packages for multi-criteria resource allocation are analyzed: Equity, HiPriority, Logical Decisions Portfolio and Expert Choice Resource Aligner. The key technical distinction concerns the type of resource allocation procedure used: Equity uses the benefit-to-cost ratio approach, HiPriority also uses the benefit-to-cost ratio approach and an exhaustive enumeration approach, whereas Logical Decisions Portfolio and Expert Choice Resource Aligner use a mathematical programming approach.

Index Terms—Multi-criteria resource allocation software, portfolio decision-making.

I. INTRODUCTION

ALLOCATING resources to projects (or programs or strategies) when not enough resources exist for them all is a demanding decision-making problem that requires balancing multiple benefits against costs, even if there are no other constraints (e.g. legal, strategic, operational) and no cost or benefit interdependencies. Suppose that a manager has n projects but he has not enough money to fund them all. Each project is indivisible (i.e. it cannot be partially financed) and can be characterized by its performance in each one of a set of benefit criteria. Suppose that the manager has assigned value scores to the performances of each project and weights to the criteria and an overall benefit score for each project was obtained by weighted summation of its value scores. How can we help the manager to find the subset of projects (portfolio) that maximizes the overall benefit given the budget constraint?

João Carlos Lourenço and Carlos A. Bana e Costa were supported by FCT (Portuguese National Science Foundation) under project PTDC/GES/73853/2006. The collaboration between the Portuguese and British authors was supported by the Portuguese Council of University Rectors and the British Council through the Treaty of Windsor Anglo-Portuguese Joint Research Programme 2008/2009.

João Carlos Lourenço is with the Centre for Management Studies of IST, Technical University of Lisbon, Av. Rovisco Pais, 1049-001 Lisbon, Portugal (corresponding author to provide phone: +351 214233523; fax: +351 214233568; e-mail: joao.lourenco@ist.utl.pt).

Carlos A. Bana e Costa is with the Centre for Management Studies of IST, Technical University of Lisbon, Av. Rovisco Pais, 1049-001 Lisbon, Portugal and with the Department of Management - Operational Research Group, London School of Economics, Houghton Street, London WC2A 2AE, UK (e-mail: carlosbana@ist.utl.pt).

Alec Morton is with the Department of Management - Operational Research Group, London School of Economics, Houghton Street, London WC2A 2AE, UK (e-mail: a.morton@lse.ac.uk).

The number of portfolios that can be formed with n projects is 2^n , ranging from the *do nothing* portfolio (no projects are funded and no benefits are realized) to the complete portfolio (which would require all projects to be funded). An exhaustive enumeration approach would consist in comparing all portfolios but it becomes impractical even if the number of projects is not too big (e.g. 10 projects will give 1024 portfolios). An alternative common sense approach consists in computing the benefit-to-cost ratio of each project and prioritizing them in decreasing order and proceed down the priority order until the budget is exhausted [1]. A third approach consists in formulating and solving a (knapsack) mathematical programming model in order to maximize the overall benefit without exceeding the budget constraint [2, 3]. Although these two approaches may seem very different they are in fact connected: as demonstrated by Dantzig [4], the portfolio given by the benefit-to-cost ratio approach is included in the optimal solution of a (relaxed) knapsack problem in which the projects are assumed to be divisible.

The benefit-to-cost ratio approach is implemented in Equity and HiPriority packages; HiPriority was inspired by Equity but it additionally has the facility to exhaustively enumerate all the possible portfolios. Alternatively, Expert Choice Resource Aligner (ECRA) and Logical Decisions Portfolio (LDP) follow the mathematical programming approach. None of these four software packages for multi-criteria resource allocation uses both approaches.

In order to help our manager we first need to study each package; this is the objective of this paper. We analyze Equity in Section III, HiPriority in Section IV, LDP in Section V, and ECRA in Section VI. Before that, we introduce in Section II some concepts that will be useful in the ensuing discussion.

II. EFFICIENCY AND CONVEX EFFICIENCY

A central concept used by all approaches is that of efficiency. A portfolio is efficient when there is no other portfolio giving more overall benefit without spending more money. We can illustrate this with reference to Fig. 1, which shows all portfolios that can be formed with six projects in the cumulative cost / cumulative benefit space. The efficient (or non-dominated) portfolios are shown as triangles in Fig. 1 and form the efficient frontier (associated for simplicity with the dotted line in the Fig. 1), whereas the dashed line links only the convex efficient portfolios (the convex efficient frontier). As can be seen there are five efficient portfolios that are not

convex efficient (those that belong to the dotted line but not to the dashed line; they are dominated by linear combinations of two convex efficient portfolios). All the efficient portfolios can be found by mathematical programming, whereas the benefit-to-cost ratio approach is limited to the convex efficient frontier.

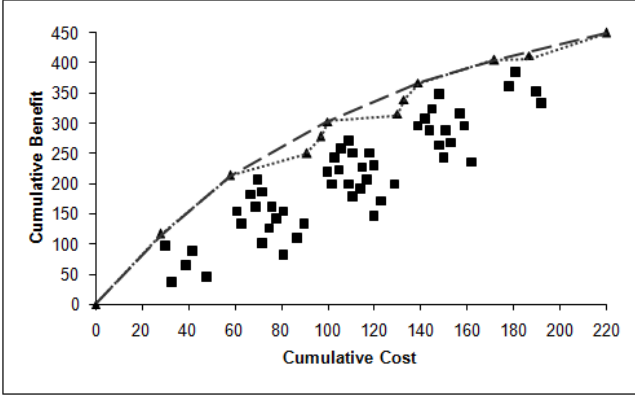


Fig. 1. Chart showing all the portfolios that can be formed with six specific projects ($2^6=64$) and the respective efficient and convex efficient frontiers.

III. EQUITY

A. Overview

Equity is a software package developed by Catalyze Ltd. (<http://www.catalyze.co.uk/products/equity>) and Enterprise LSE Ltd. (<http://www.lse.ac.uk/collections/enterpriseLSE/>) and herein we analyze its version 3.4. Equity includes features that allow scoring projects in each criterion, weighting the criteria, determining additively the overall benefit score for each project, and to find the convex efficient portfolios.

Equity is designed to organize projects by areas (e.g. regional areas, business units or departments). An area is represented by a column and a project by a box inside a column (see Fig. 2). An area can be *cumulative*, if multiple projects in the area can be chosen; or *mutually exclusive* if only one can be chosen. A model with mutually exclusive areas implies an underlying multiple choice knapsack problem in the sense of Zemel [5]. (See examples of both types of area in Fig. 2.)

The usage philosophy behind Equity is that each area of an organization should assign benefit scores and costs to its own projects. However, in the point of view of the company, the ranges between the best and the worst projects performances in one criterion for all areas are not usually worth the same; Equity overcome those differences by converting the area scales to a scale common to all areas using within-criterion weights, which are assessed through a swing weighting procedure. The across-criteria weights, which are assessed through a swing weighting procedure, are used to transform the scores of the projects in each criterion in overall scores.

B. Approach

Equity uses the benefit-to-cost ratio to find the convex efficient portfolios along the admissible range of cost and

show them in a chart (see Fig. 3). If the user decides to choose a portfolio P (see Fig. 3) that is not on the convex efficient frontier, Equity would recommend two convex efficient portfolios: a *cheaper* portfolio and a *better* portfolio (respectively denoted by C and B in Fig. 3). The user can also set a budget line in which case Equity will identify the two closest portfolios on the convex efficient frontier on either side of its intersection with the budget line. (See [1] for details.)

The software allows users to navigate the portfolio space interactively, using both the cityscape (Fig. 2) and frontier (Fig. 3) displays. Equity allows *must have* projects to be included in the portfolios and provides a feature which allows users to *force in* projects: when a project is forced in, its cost is subtracted from the overall resource budget, giving users an immediate sense of the marginal projects they would have to give up if they were to accept the forced in project.

C. Additional features

With Equity is possible to explore the convex efficient frontier for several budgets. It is also possible to perform sensitivity and what-if analysis by directly changing the parameters (costs, benefit scores and weights) of the model. Additionally, Equity allows risk-adjusting the scores of the projects by assigning probabilities to each criterion. The adjustment is done by: converting the probability to a penalty criterion if the option *risk adjusted (negative benefit)* is selected; or by multiplying the probability with the benefit

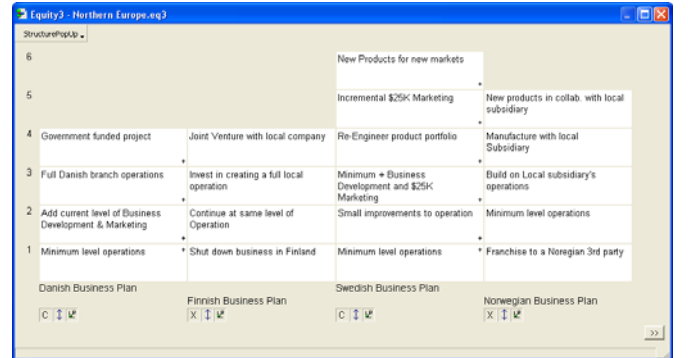


Fig. 2. Equity screenshot showing four areas (columns) and their respective projects of the model *Northern Europe* (bundled with the software). A cumulative area is identified by a C below the area name (see e.g. the *Danish Business Plan* area) and an exclusive area by an X (see e.g. the *Finnish Business Plan* area).

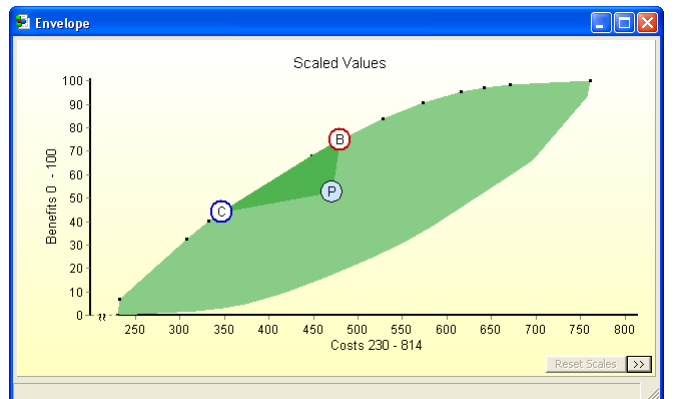


Fig. 3. Equity screenshot showing the portfolios convex envelope of the model *Northern Europe*. The convex efficient portfolios are marked on the upper line with dots.

score if the option *risk adjusted (multiplicative benefit)* is selected.

IV. HiPRIORITY

A. Overview

HiPriority is a software package developed by Krysalis Ltd. (<http://www.krysalis.co.uk/>) and herein we analyze its version 3. HiPriority includes features that allow to score the projects in each criterion, to weight the criteria, and to determine the overall benefit score and the costs for each project.

When HiPriority starts running shows its *process diagram* window with the intent to guide the user in the decision analysis process. The project scores in each criterion are determined in HiPriority through *mappings*. Criteria weights can be assessed using either the trade-off procedure or the swing weighting procedure.

B. Approach

HiPriority uses the benefit-to-cost ratio to find the convex efficient frontier (see the top line in the chart in Fig. 4) and uses exhaustive enumeration to find the efficient frontier.

Buffering in or buffering out a project is a way to respectively include or exclude a project from the portfolios. With HiPriority more constraints can be added through: (1) *Dependencies* between any two projects (e.g. project A is only included in the portfolio if project B is also included). Indeed, it is possible to define bi-directional dependencies between two projects in case they are mutually inclusive (e.g. project A is only included in the portfolio if project B is also included and vice-versa). The dependency relationship can be extended to more than two projects by transitivity (e.g. project A is only included in the portfolio if project B is also included and vice-versa, and the same applies between projects B and C, hence A, B and C are mutually inclusive projects). (2) *Exclusions* between any two projects to define mutually exclusive relationships (e.g. either project A or project B could be in the

portfolio but not both). Because there are no restrictions in the number of exclusions that a project may have with other projects, it is possible to define mutually exclusive relationship in sets of more than two projects.

C. Additional features

HiPriority allows to define synergies between two projects through *modifiers* (see Fig. 5), when the benefit (or the cost) of both projects is different from the sum of their benefits (or costs). With HiPriority is possible to explore the convex efficient frontier for several budgets. It is also possible to define several *weight sets* that could be used to test how the portfolios differ when different criteria weights are used. When HiPriority is not able to determine the efficient frontier due to the size and/or complexity of the model a *walk* could be used; a walk is a sequence of portfolios that starts from a chosen portfolio and, at each step of the walk, creates a new portfolio by adding from the remaining admissible projects the one with the best ratio benefit-to-cost.

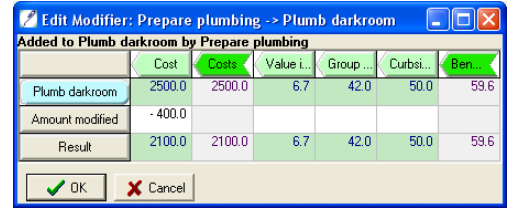


Fig. 5. HiPriority screenshot using the model *House Renovation* (bundled with the software). It shows a modifier established for the project *Plumb darkroom* to be applied if the project *Prepare plumbing* is to be present in the portfolio, which would imply a reduction in the Cost from 2500 to 2100.

V. LOGICAL DECISIONS PORTFOLIO

A. Overview

Logical Decisions Portfolio (LDP) is a Microsoft Excel add-in that takes as input the overall benefit scores of the projects from a Logical Decisions for Windows (LDW) model and, if wanted, the cost and resource consumption data for each project. LDP and LDW were developed by Logical Decisions (<http://www.logicaldecisions.com/>) and herein we analyze the LDP version 1.110.

LDW implements several methods to assess scores (Single Measure-Utility Function (SUF), Analytic Hierarchy Process (AHP) [6], AHP SUF and Adjusted AHP) as well as several criteria weighting techniques (Tradeoff procedure, SMART, SMARTER, Pairwise Weight Ratios, AHP). It uses a weighted sum to obtain the overall benefit scores of the projects. (More details about LDW are presented on [7].)

B. Approach

LDP uses the mathematical programming approach to find the best portfolio of projects that meet the constraints (see Fig. 6).

LDP assumes that the projects benefit scores which come from LDW are expressed in interval scales and, therefore, cannot be used in the mathematical programming model without being first transformed to ratio scales. LDP has two ways of doing this transformation (see the *scale benefits* edit

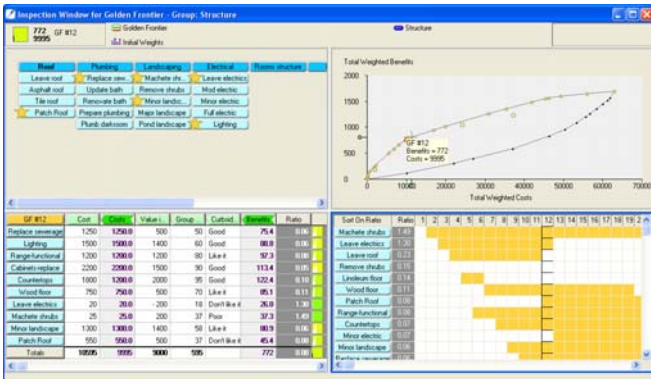


Fig. 4. HiPriority screenshot showing an inspection window with the convex efficient portfolio #12 selected, using the model *House Renovation* (bundled with the software). The top-left window shows the projects grouped in columns; the projects of the portfolio #12 are signaled with stars. The top-right window shows the convex envelope of the portfolios. The bottom-left window shows for each project in portfolio #12 their respective data. The bottom-right window shows the projects ranked by their benefit-to-cost ratios highlighting those present in convex efficient portfolios.

boxes labels in Fig. 7): it asks the decision maker to state a benefit score x in such a way that one project with benefit 1.0 is worth the same as two projects with benefit x or; it asks the decision maker to state a benefit score y that is worth zero dollars. With either one of these values entered LDP performs a benefit scale transformation. (See the LDP on-line help for more details about these transformations).

There are several types of constraints that could be added to a LDP model: *budget constraints* for several periods of time, inclusion or exclusion of a project, *if-then constraints* to establish dependencies between two projects (e.g. project A is included only if project B is also included; or project A is included only if project B is not included).

By defining *groups* of projects it is also possible to define *group constraints* that allow: to include in the portfolio, or exclude from it, all the projects of a group; to include either one or none projects of a group in the portfolio; to include either all projects or none of the projects of a group in the portfolio; to include exactly one of the projects of a group in the portfolio; to include at least one of the projects of a group in the portfolio.

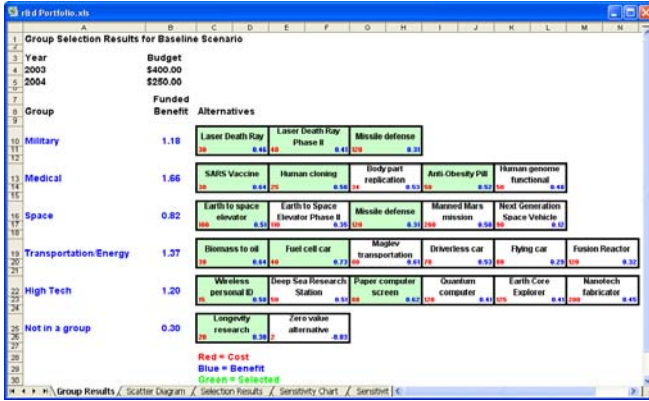


Fig. 6. LDP screenshot showing the *Group Results* for the *Baseline Scenario* of the model *R&D Options* (bundled with the software). Each group of projects (*Military*, *Medical*, etc.) and the projects not included in any group (*Not in a group*) are shown in rows. The selected projects have a green background.

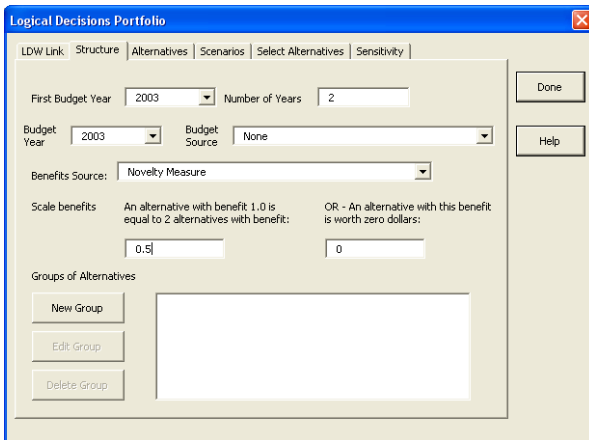


Fig. 7. LDP screenshot showing the *Structure* window of the model *R&D Options* (bundled with the software). Note that in order to transform the benefits scale (in *Scale benefits*) was entered 0.5 in the edit box *An alternative with benefit 1.0 is equal to 2 alternatives with benefit*.

C. Additional features

LDP allows one to add *resource constraints* to ensure that the projects included in the portfolio do not use more resource than is available (with a less than or equal constraint) or that they use at least a defined amount of a resource (with a greater than or equal constraint). It is also possible to add allocation constraints that specify a minimum percentage of the budget that must be spent on projects belonging to a particular group.

By using a *sensitivity model* one can run several mathematical programming instances using different amounts as budget. The budget used in each run is incremented by a defined amount between a starting and ending budgets; the increment and the starting and ending amounts are defined by the user.

Finally, it is also possible to establish different *scenarios* in LDP in order to obtain portfolios using different budgets and constraints. A scenario can use either the *requested budgets* or the *allowable budgets* of the projects (an allowable budget for a project is necessarily smaller than the requested budget and usually implies a reduced benefit).

VI. EXPERT CHOICE RESOURCE ALIGNER

A. Overview

Expert Choice Resource Aligner (ECRA) is a piece of software that extends the capabilities of Expert Choice (EC) to resource allocation. This software was developed by Expert Choice Inc. (<http://www.expertchoice.com>) and is available jointly with EC given that a serial number to unlock its features is entered during the setup process. Herein we analyze the version 11.5 of this software package.

EC is a software implementation of the AHP [6], a pairwise comparison approach which lets one determine scores for the projects on each criterion, weights for the criteria and overall benefit scores for the projects. (More details about Expert Choice are presented on [7].)

B. Approach

ECRA uses the mathematical programming approach to find an efficient portfolio for a certain budget (see the field *Budget Limit* near the top-left corner in Fig. 8) or to find a set of efficient portfolios using different budget levels (see the *efficient frontier* chart in Fig. 9). The user is free to choose a higher or lower number of budget levels.

Certain pre-defined types of constraints can be added to a ECRA model, namely *Dependencies* between two projects (here exemplified by projects A and B): dependency (project A is only included in the portfolio if project B is also included), mutual dependency (projects A and B are both in the portfolio or neither of them are), mutual exclusivity (either project A or project B could be in the portfolio). It is also possible to compulsory include a project in the portfolio or to preclude a project to be considered for a portfolio. *Groups* of projects can be defined in such a way that the number of the projects selected for the portfolio from the group has to be: less than or equal to one, equal to one, at least one.

C. Additional features

It is possible to add additional pre-defined types of constraints using the initial window of ECRA (shown in Fig. 8). E.g. in the bottom-left area of the window one can use the columns: *partial*, to allow for partial funding of a project (what implies that the underlying variable becomes real instead of being integer); *min %*, to define a minimum percentage of funding for a project. The bottom-right area of the initial window can be used to add other constraints. Fig. 8 shows two columns (constraints) named *network* and *project mgrs*; each row of these columns shows the resource consumptions of the respective project; the bottom of the columns show the minimum and the maximum allowed resource consumptions (the minimum is blank in the example shown), the levels of consumption of the current portfolio (in the row *funded*) and the *total* consumption of all projects. *Funding pools* are used to define limits for the resource consumption of the projects.

With ECRA scenarios with different constraints sets (*budget limits*, *musts*, *must-nots*, *custom constraints*, *dependencies*, *funding pools* and *groups*) can be created and evaluated.

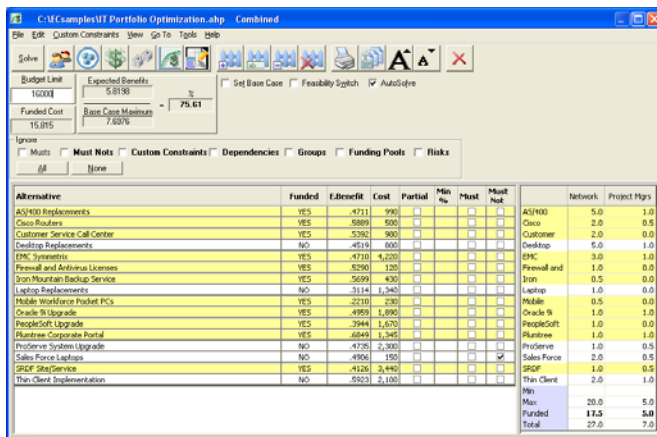


Fig. 8. Expert Choice Resource Aligner screenshot of the model *IT Portfolio Optimization* (bundled with the software) showing the portfolio of projects selected for a budget limit of 16,000 (the projects included in the portfolio are signaled with *yes* in the column *Funded*).

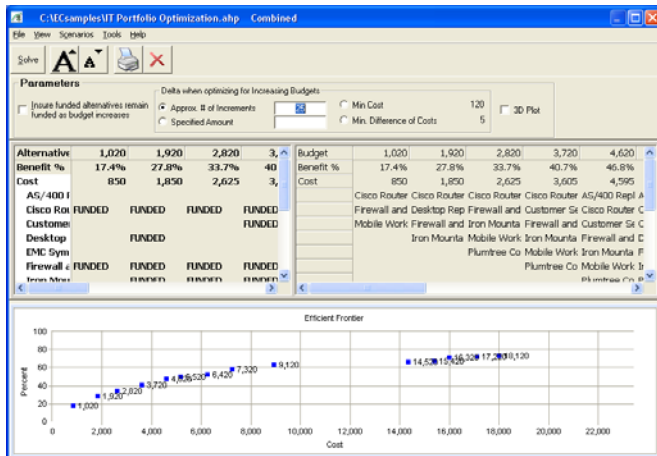


Fig. 9. Expert Choice Resource Aligner screenshot of the model *IT Portfolio Optimization* (bundled with the software) showing efficient portfolios for different budgets.

The benefit associated to a project can be “corrected” by its probability of success, which by default is equal to 1. For a certain risk r the probability of success is calculated by $1-r$ (with $0 \leq r \leq 1$), or vice-versa. The expected benefit will be equal to the probability of success times the benefit.

VII. FINAL REMARKS

A. Approaches and features

The key technical distinction between the software packages analyzed concerns the type of resource allocation procedure used: Equity uses the benefit-to-cost ratio approach; HiPriority use the benefit-to-cost ratio approach and also uses an exhaustive enumeration approach; whereas LDP and ECRA use the mathematical programming approach. Therefore only LDP and ECRA can deal with additional constraints. It would be interesting to have software that could combine the benefit-to-cost approach and the mathematical programming approach, but none of the software here analyzed has both approaches implemented. In face of the analysis done we started to develop a software package – that we called RAMS (Resource Allocation for Management and Support) – that uses the benefit-to-cost ratio and the mathematical programming approach, and could therefore handle additional constraints.

With Equity, HiPriority and ECRA it is possible to view a chart with the efficient portfolios and explore it for different budgets, which is an appealing feature because, as Kleinmuntz said [8], “In most organizations, this is a ‘soft’ constraint, since the Chief Financial Officer has some degree of discretion to increase or decrease capital spending”. However, Equity only shows the convex efficient frontier; HiPriority shows the convex efficient frontier and, if the number of projects is small, the efficient frontier; and ECRA shows an “efficient frontier” that does not guarantee the inclusion of all the efficient portfolios (the number of efficient portfolios that can be found depends heavily on the budget levels defined by the user).

HiPriority is the only software package that allows including (positive or negative) synergies between interdependent projects.

B. Problems

Hereafter we present a problem related with a procedure implemented by two pieces of software and another problem concerning the (lack of a) definition of a *true zero* in the benefits scale used:

- 1) AHP is a procedure that suffers from several reported flaws (see e.g. [9, 10]). Because the portfolios obtained by ECRA depend on the input values it receives from EC, and the latter is an implementation of AHP, the results of ECRA also suffers from those flaws. The same problem occurs with LDP whenever AHP is the procedure used in LDW.
- 2) The portfolio analysis requires the use of ratio scales in both cost and benefit. The cost of the projects is usually measured in money, which is a ratio scale; although, benefit scales usually have arbitrary zeros and hence are

interval scales (as is the case with the Fahrenheit and the Celsius temperature scales, whose zeros do not indicate the absence of temperature). (For information concerning measurement scales see e.g. [11].) However, this problem can be addressed in two ways: (i) the procedure used for assessing scores in each criterion can be changed to produce ratio scales instead of interval scales (e.g. by using difference judgments over a *do nothing* project: the difference between the project *do nothing* and itself is a true zero, and the difference between the project *do nothing* and the remaining projects would be greater than or equal to zero); (ii) An interval scale can be transformed into a ratio scale by defining a true zero upon it. The only packages of software herein analyzed that address this concern are Equity, which allows to automatically include a *do nothing* project to areas with cumulative projects, and LDP, which asks for a parameter to operate a scale transformation.

REFERENCES

- [1] L. D. Phillips and C. A. Bana e Costa, "Transparent prioritisation, budgeting and resource allocation with multi-criteria decision analysis and decision conferencing," *Annals of Operations Research*, vol. 154, pp. 51-68, 2007.
- [2] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*. Chichester: John Wiley & Sons, 1990.
- [3] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Berlin: Springer, 2004.
- [4] G. B. Dantzig, "Discrete-variable extremum problems," *Operations Research*, vol. 5, pp. 266-277, 1957.
- [5] E. Zemel, "The linear multiple choice knapsack problem," *Operations Research*, vol. 28, pp. 1412-1423, 1980.
- [6] T. L. Saaty, *Multicriteria Decision Making: The Analytic Hierarchy Process* vol. 1. Pittsburgh, PA: RWS Publications, 1990.
- [7] S. French and D.-L. Xu, "Comparison study of multi-attribute decision analytic software," *Journal of Multi-Criteria Decision Analysis*, vol. 13, pp. 65-80, 2005.
- [8] D. N. Kleinmuntz, "Resource allocation decisions," in *Advances in Decision Analysis: From Foundations to Applications*, W. Edwards, R. F. Miles Jr., and D. von Winterfeldt, Eds. Cambridge: Cambridge University Press, 2007, pp. 400-418.
- [9] J. S. Dyer, "Remarks on the analytic hierarchy process," *Management Science*, vol. 36, pp. 249-258, 1990.
- [10] C. A. Bana e Costa and J.-C. Vansnick, "A critical analysis of the eigenvalue method used to derive priorities in AHP," *European Journal of Operational Research*, vol. 187, pp. 1422-1428, 2008.
- [11] J. C. Nunnally and I. H. Bernstein, *Psychometric Theory*, 3rd ed. New York: McGraw-Hill, 1994.