

INSTITUTO DE COMPUTAÇÃO

UNIVERSIDADE ESTADUAL DE CAMPINAS

A Project Portfolio Selection Problem

Cleber Mira *Pedro Feijão*
Maria Angélica Souza *Arnaldo Moura*
João Meidanis *Gabriel Lima*
Rafael Schmitz *Renato P. Bossolan*

Technical Report - IC-12-21 - Relatório Técnico

June - 2012 - Junho

The contents of this report are the sole responsibility of the authors.
O conteúdo do presente relatório é de única responsabilidade dos autores.

A Project Portfolio Selection Problem

Cleber Mira¹, Pedro Feijão^{1,2}, Maria Angélica Souza¹, Arnaldo Moura², João Meidanis^{1,2}, Gabriel A. Costa Lima⁴, Rafael Schmitz³
and Renato P. Bossolan³

¹Scylla Bioinformatics

²Institute of Computing, Unicamp

³AES-Tietê

⁴Mechanical Engineering Department, Unicamp

Contents

1	Introduction	2
2	PPS and Input Instances	3
2.1	Input Data	5
3	A PPS Model	10
3.1	Decision variables	10
3.2	Objective function	11
3.3	Constraints	12
4	A Heuristics Solution	14
4.1	The GRASP Metaheuristic	14
4.2	A GRASP variant for PPS: kCRGH	15
5	Experiments	19
6	A Decision Support System Prototype	22
7	Conclusions	30

8 Acknowledgment

31

Abstract

The *project portfolio selection* (PPS) problem consists of constructing a project portfolio, that is a selection of projects scheduled over a period of time using various, potentially conflicting criteria and resource constraints. The PPS is a well known problem, recurrently occurring in several industries, with a rich history of approaches for modeling it and a large number of techniques for solving it.

In this work we present a model for the PPS problem based on a real-world problem of selection and scheduling projects stemming from the electricity generation industry. This model includes constraints that better reflect a particular kind of interdependence among projects. We also propose a heuristics based on the metaheuristics GRASP, to solve the problem and assess its quality and performance through computational experiments. We describe the implementation of a decision support system prototype for the PPS problem that realizes the proposed heuristics and includes several features that can help decision makers through the project portfolio selection process.

1 Introduction

Decision makers are usually confronted by the problem of choosing from a large number of projects a subset of those, that is, a project portfolio, that would prove to be the more beneficial according to a combination of several criteria and given a limited amount of resources. The construction of a project portfolio involves the selection and scheduling of projects over a period of time and using various, potentially conflicting criteria and resource constraints. This is known as the *project portfolio selection* (PPS) problem. It is possible to define the problem in different ways depending on the characteristics that will be modeled.

Most of the works concerning the PPS problem provide solutions for simulations and randomly generated data. Instead, here we model a real-world PPS and present solutions for real instances of the problem. Such real-world instances may contain thousands of projects and their PH may span a few dozens of periods.

In this work, we consider that a project is not divisible into smaller activities. We also model the idea of a *planning horizon* (PH), that is, a sequence of time periods with a predefined amount of resources available at each time period. Each project should be scheduled at a starting time and should allocate its share of available resources according to its duration. Projects may also be interdependent on each other in several ways. For instance, a project starting time may be dependent on another project's ending time. This work also presents a new model for the PPS problem that includes constraints that better reflect the interdependence among projects. The input data

stems from a real-world problem of selection and scheduling projects that often occurs in the electricity generation industry. We propose a heuristics to solve the problem and implement a computational experiment to compare the quality of solutions and the performance of the heuristics when applied to randomly generated instances and also to an actual instance extracted from a real scenario.

We review the PPS problem and the real-world input instance in Section 2. The particular model for the PPS problem treated here is detailed in Section 3. Section 4 presents a greedy heuristics for solving the PPS problem based on the GRASP meta-heuristics. The computational experiments are discussed in Section 5. In Section 6 we describe a prototype for a decision making support system that implements the heuristics and offers features that support the analysis and construction of portfolios. Finally, in Section 7 we summarize the results and discuss new improvements.

2 PPS and Input Instances

There are several variant formulations for the PPS problem [7] depending on specific constraints, objective functions, decision variables, resource availability policies, and time assumptions. However, these variants usually share some characteristics, and so a PPS problem can be summarized as follows: find an instance of a project portfolio, that is, a selection and a schedule of projects along time periods, that maximizes an objective function over all viable portfolios. The general conditions involved in the PPS problem are thus:

1. a sequence of time periods (PH);
2. an assignment of available resource values to each time period;
3. a set of projects;
4. a benefit function that measures how beneficial a project is when executed.
5. a cost function that describes a project demand for resources along its duration.
6. a set of constraints representing different properties of the problem, such as resource limitation, project interdependencies, and scheduling restrictions; and
7. an objective function that measures how beneficial a portfolio is.

Some of the most common differences found in PPS variants are based on the following characteristics [7]:

Selection only vs. selection and scheduling Problems that involve the selection of problems only, and so no PH need to be defined.

Transference vs. non transference of remaining resources There are project portfolio instances in which the total of resources demanded by the projects scheduled to a particular time period is inferior to the amount available at that time period. Instead of potentially wasting the excess of resources at each time period, it is possible to define a PPS variant that allows the transference of the remaining unused resources along the subsequent time periods.

Independency vs. interdependency of projects PPS variants may consider various dependence relations among projects.

Interdependency among projects is a particular fruitful source of PPS variants. Projects may be interdependent in several ways [21, 15], such as by precedence, complementarity, incompatibility, or synergy, depending on the way they share resources and provide benefits when executed along the PH. The following are typical examples:

1. dependencies based on precedence. For instance, a project j dependent on a project i by precedence must be scheduled to a time period later than the ending time of i .
2. time restrictions based dependencies. Such as the obligation of starting a dependent project at a specified number of time periods before or after the ending time of the project it depends on. For instance, project i has to start r time periods before or after the ending time of project j .
3. dependencies based on synergy. Synergetic projects are favored to be chosen together, but there is no obligation that all sinergetic projects be selected, since their scheduling depends on how much beneficial is each project and on the benefit of a set of sinergetic projects with respect to the criterion functions.
4. dependency among projects based on complementarity. That is, projects that should be scheduled together, at the same time period. It may seem that they should be modeled as a single project, but complementary projects might contribute differently to each objective function and their scheduling might be influenced by the decision-maker preferences. For this reason, by modeling complementary projects as distinct projects, we give the decision-maker a better picture of how her preferences may impact the project scheduling [1].
5. dependency based on incompatibility, i.e. projects that are mutually exclusive. In other terms, by choosing a project, some other projects could not be chosen.

Several methods for solving PPS variants have been proposed in the literature [4, 6, 5, 13]. “Benefit” measurement techniques include economic models [30, 25], decision theory models [26, 23, 32], e.g. decision tables and decision trees, and comparative methods, e.g. the analytical hierarchy process [41], scoring methods [27, 31],

and multi-attribute utility theory [2, 17]. “Benefit” measurement techniques have the advantage of using qualitative and quantitative aspects of both projects and decision parameters to find a portfolio. However, these methods present limitations since they are not able to handle adequately multiple constraints and project interdependency. Mathematical programming techniques yield solutions involving single objective programming [21], multiobjective programming [40, 39, 29, 18], and goal programming [9, 10]. Other techniques may involve various approaches such as cluster analysis [33], data envelopment analysis [11] (DEA), fuzzy techniques [8], and metaheuristic algorithms [24]. Metaheuristic algorithms usually perform well when applied to real world instances, even though they do not guarantee optimal solution quality. Among metaheuristic methods that offer a good trade-off between performance and quality, we can mention the analytic network process [34, 35], genetic algorithm based heuristics [28], ant colony optimization [16], simulated annealing [14], and greedy methods [38, 19].

2.1 Input Data

AES-Tietê is a leading energy generation company whose mission is to generate and sell energy by safer and sustainable means to attend customers, communities, collaborators, suppliers, and society needs. To achieve such a vision, AES-Tietê managers are responsible for the yearly task of selecting and scheduling projects over a period of time by assessing their costs, controlled risks, and the company available resources; i.e. there is the yearly task of constructing a project portfolio. We describe here the characteristics and assumptions about the input data available to decision makers for building a portfolio at AES-Tietê.

Planning Horizon The PH usually takes 5 years; that is, it is composed by 60 monthly time periods. When visualizing data, the first two years are presented as 24 monthly fields, together with 5 fields representing the total yearly project costs and available resources.

Available Resources The amount of available resources are specified by the decision makers following the management strategical directions.

Projects The input data initially provided by AES-Tietê was composed by a set of projects, each project comprising of several parameters. Projects were arranged in a spreadsheet table where several rows may contain data for a single project and each column contains a specific project parameter value. For practical reasons, we worked together with AES-Tietê management and redesigned the spreadsheet format to a new standard, so that project parameters for a single project are displayed in a single row in the table. We also suggested the inclusion of new columns holding

relevant information required for building a portfolio, such as fields for representing project dependencies and mandatory projects, which were not previously covered in the initial spreadsheet format. The columns in the standard spreadsheet format are the following:

1. Project number: a unique, numeric identifier assigned to each project that is included in the portfolio.
2. Classification: projects can be classified into two categories according to the kind of cost they incur: operational expenditures (OPEX) or capital expenditures (CAPEX).
3. Area: the area might be a name or nickname for a power plant, or a general area in the company, where the project belongs to.
4. Description: a descriptive name for the project.
5. Total controlled risk: a value that indicates how much risk is avoided by a project in case it is selected and executed, which means that the greater the controlled risk the larger the amount of risk avoided.
6. Dependency identifier: the number of the project which the current project depends upon.
7. Dependency months: the number of shift months from a dependent project, that is, the number of months before or after the ending time of the project the current project depends upon. This parameter can be negative, zero (null is also accepted), or a positive numeric value. Consider a project i dependent on another project j by $-a$ months, where a is a positive integer. In this case i have to start a months before the end of project j or later. If i is dependent on j by a months then i should start a months after j ending time or later.
8. Lead time: the number of time periods after the beginning of the PH when a project is allowed to be scheduled and its costs start to consume resources. When filled with a blank value the lead time is assumed to be zero.
9. Costs: a project cost is distributed among 24 columns corresponding to the months of the first two years of the PH. These columns are followed by 5 columns containing the yearly total cost of a project, one column per each year of the PH.
10. Mandatory: this column indicates whether a project is mandatory. If a project is mandatory then it must be scheduled at the first time period where it indicates a non-zero cost.

Proj. number	Classification	Area	Descr.	Risk	Dep. Id	Dep. Month	Lead Time
1	OPEX	A1	p1	21.6			
2	CAPEX	A2	p2	0.25			
3	CAPEX	A2	p3	3.87	2	-3	
4	OPEX	A3	p4	5.68			
5	OPEX	A4	p4	5.68			

Table 1: Schematic example of a spreadsheet containing an input instance. The table depicts the project number, classification, area, description, total controlled risk, dependent id, dependency month, and lead time columns. Each row contains parameter values for a single project. Cost columns are found in the Table 2.

Proj. number	...	First month cost	...	First year cost	...	Last year cost
1	...	20000	...	340000	...	250000
2	...	6000	...	180000	...	360000
3	...	15500	...	220000	...	727000
4	...	50000	...	850000	...	680000
5	...	70000	...	1200000	...	970000

Table 2: Schematic example of the spreadsheet of an input instance showing cost columns. Each row contains parameter values for a single project. We omitted cost columns with similar kind of data.

Assuming that the first time period with non-zero cost indicates a project *formal initial time* and the last time period with non-zero cost indicates a project *formal ending time* then the input data spreadsheet defines not only the set of restrictions on the data, but also a portfolio.

Table 1 and Table 2 show a brief schematic view of how projects and costs are presented in a spreadsheet.

In our experiments, we used a spreadsheet containing a real data instance, with 2349 projects, whose total cost along 5 years is around 479 millions (BRL). These projects were proposed by the company technicians in 2011, in order to plan the project portfolio from 2012 through 2016. Projects in this instance are distributed among 20 areas. The major amount of projects (1906) are classified as OPEX; the remaining (443) are classified as CAPEX.

The input instance found in the spreadsheet not only conveys information on a project duration and its demand for resources along its duration, but also it contains a

schedule of each project, obtained when a project start time is defined as the first non zero cost field in a project row. Therefore the spreadsheet presents a portfolio that can be understood as a manual solution for the PPS problem. We call the portfolio described by the spreadsheet the *initial portfolio*. Even though the initial portfolio is a valid solution for the PPS problem, we do not take it into consideration. Instead, we use the essential data on projects and their parameters, ignoring the initial portfolio schedule during the optimization process.

After a series of meetings involving managers and decision makers, we identified the best set of restrictions used by management to define an adequate portfolio. The restrictions considered in this real case were:

1. Eventual scheduling: a project will be eventually scheduled at a time period in the PH, not beyond PH boundaries. The last time period of the PH has no constraint restricting a project cost.
2. Mandatory projects: certain projects should start at a specified time period.
3. Lead time: a project may take some time of preparation before being definitely scheduled and can start to consume resources. A project lead time indicates how many time periods since the beginning of the PH the project will be ready to be scheduled. Therefore, a project should not start before its lead time.
4. Project dependency: a project i may be dependent to another project j , meaning that project i should be scheduled a certain number of time periods after or before project j ending time. If the sum of j ending time and the number of dependent time periods results in a time period outside of the PH time scale, then the project is not selected, that is it is not included in the portfolio.
5. Total costs per year: the total sum of project costs per year cannot surpass the amount of resources available at that year, even if project costs are monthly specified. The partition of projects between CAPEX and OPEX reflects a division among projects related to investments and projects related to operational costs only. The amount of available resources is divided between these two categories and so impose the restriction that the sum of project costs of a certain category cannot surpass the total amount of resources available for that category, for each year.

Objective function One of the main tasks during the modeling of the PPS problem was to find which objective function and possible constraints would better represent the problem at hand. We tested a few alternatives for a single objective function. We omit here some of the developments involved at reaching an adequate objective function and focus instead on the parameters and problem properties that have remained significant. The main parameter used for optimization purposes is the total

controlled risk. One should want to find a portfolio that maximizes the amount of total controlled risk without violating any constraint.

The first attempt at modeling the objective function was to maximize the sum of total controlled risks of all projects scheduled during the PH. However, we found a difficulty with this objective function: the function invariability, that is, the objective function has the same value for any portfolio containing the same projects, independently of which time period the project was scheduled at in the solution. Since the sum of total controlled risks is the same for any portfolio containing the same set of projects, it is not possible to compare solutions using this value only.

The invariability of the sum of total controlled risks could be tolerated if somehow we could include in the objective function the information about when a project was scheduled. In other terms, the objective function should consider not only the controlled risks, but also the various ways of scheduling projects in a portfolio. By capturing ways of scheduling projects, the objective function could provide a representative value that could be used to compare solutions. We obtained a better objective function by considering the area under the cumulative controlled risk function, along the PH.



Figure 1: Area chart of cumulative total controlled risk of two distinct portfolios. The cumulative total controlled risk of portfolio *A* is the area under the dashed curve, while the area under the solid curve represents the cumulative total controlled risk of portfolio *B*. Portfolio *B* controls a larger amount of risk earlier than portfolio *A*.

Each portfolio defines a function that returns the sum of total controlled risks of all projects until a given time period of the PH. Portfolios in which projects are scheduled earlier in the PH will have a larger area under the curve than the area of portfolios in which projects are scheduled later in the PH. Figure 1 illustrates the different areas covered by the curves obtained from two portfolios, *A* and *B*, for the same PPS problem. The cumulative total controlled risk of portfolio *A* is the area under the dashed curve, while the area under the solid curve represents the cumulative

total controlled risk of portfolio B . The area below the curve of portfolio B is larger than the area below the curve of portfolio A , implying that portfolio B controls a larger amount of risk earlier than portfolio A .

Another relevant aspect that have an influence on the value of the objective function is when the benefit of the controlled risk associated with a project should be taken into consideration by the objective function. In the previous discussion, we have assumed that a project total controlled risk contributes to the objective function value at the moment the project is scheduled in the PH, i.e. at the project starting time. However, decision makers might find it more adequate to consider that the benefit of controlled risks should be accounted for only at the end of a project. Thus, there are two possible ways of constructing the objective function:

- A project total controlled risk is always added to the objective function, even if the project end period is beyond the PH. This view about the contribution of controlled risks is somewhat counter-intuitive because one would expect that a risk would be completely controlled by a project only at the moment it terminates, and that should be inside the PH.
- The total controlled risk of a project is not summed up to the objective function when there is a controlled risk contribution that takes effect at a time period beyond the PH. In this case the contribution of a project controlled risk is made available at the time period immediately after the project last period of execution. As such, risks controlled by project terminating outside the PH will not impact the objective function.

We adopted in our model the second definition, that is, the total controlled risk of a project is added to the objective function at the period immediately after termination of the project, assuming this occurs inside the PH. This definition is relevant when identifying not only the best start time periods of projects, but also the impact of a project's ending time on the value of the objective function.

3 A PPS Model

We developed a model for representing a specific version of the PPS problem stemming from a real case occurring at the AES-Tietê energy generation company.

3.1 Decision variables

We consider a set of I projects from which to select a portfolio according to a set of criterion functions and resource constraints. We define a decision variable x_{it}

to indicate whether project i was chosen to start at period t , where $i = 1, \dots, I$, $t = 1, \dots, T + 1$, and T is the number of time periods in the PH, as:

$$x_{it} = \begin{cases} 1 & \text{if project } i \text{ starts at time period } t \\ 0 & \text{otherwise} \end{cases}$$

3.2 Objective function

The criteria used to assess when a project should be scheduled was its contribution to the *controlled risk* of the portfolio. In our terminology, controlling a risk involves taking measures to prevent the risk occurrence. Clearly, risks should be avoided, and, in this particular version of a PPS problem, the assessment of a risk controlled by a project stands for the project capacity of bringing benefits and managing undesirable consequences.

As a criterion that expresses the desire of maximizing the risk we considered the function of cumulative controlled risk over the PH, for all projects.



Figure 2: Cumulative Risk graph

Figure 2 shows the total cumulative controlled risk of a portfolio for a PH of 5 years. The total cumulative risk is obtained by summing all controlled risks associated to projects scheduled in each year of the PH, in a cumulative way, that is, risk values from previous years are summed up to a specific year. Notice that the project total risk is only considered as controlled at the end of the project life time. The area below the curve of Figure 2 is expressed by

$$\sum_{i=1}^I \sum_{t=1}^T (T - t - d(i) + 1) R_i x_{it} \quad (1)$$

where $d(i)$ is the duration, R_i is the risk controlled by project i .

We can use (1) as the objective function for the problem. It has the explicit advantages of being easily computed and of being expressible by a chart that allows a simple comparison between solutions.

3.3 Constraints

Eventual scheduling We consider that every project will eventually start at some time period in the PH and for this reason the following constraint holds:

$$\sum_{t=1}^T x_{it} \leq 1 \quad \text{for } i = 1, \dots, I \quad (2)$$

We adopt by convention that the time period $(T + 1)$ represents a time slot when every project that could not be scheduled earlier in the PH should be assigned to.

Mandatory projects There are also *mandatory projects*, that is projects that should start at some time period before the last time period in the PH. We present here the constraints involving mandatory projects for completeness. In fact, the heuristics deals with mandatory projects by scheduling them and subtracting their costs from the available resources, before any optimization procedure. Let M be the set of mandatory projects and, for $i \in M$, we denote by $itp(i)$ the initial time period of project i , that is, the period at which i was scheduled as defined in the input instance. The value of $itp(i)$ can be obtained by the equation:

$$itp(i) = \sum_{t=1}^T tx_{it} \quad \text{for } i = 1, \dots, I, \quad (3)$$

where x_{it} is fixed a priori when i is a mandatory project. Thus, the following holds:

$$x_{i,itp(i)} = 1 \quad \text{for } i \in M. \quad (4)$$

Lead time Some projects need some preparation time before being available for scheduling. These projects should be scheduled at certain time period t equal to or later than a time period defined for the project, called its *lead time*. The lead time value of a project i is the number of time periods after the beginning of the PH when i is allowed to be scheduled at. We assume that the lead time is inside the PH. Denoting the lead time of a project i by $lt(i)$, the following lead time restrictions hold:

$$\sum_{t=1}^{lt(i)-1} x_{i,t} = 0 \quad \text{for } i = 1, \dots, I \quad (5)$$

Project dependency We also modeled a particular kind of interdependence between projects related to the ending time period of the project one depends upon. If project j is dependent upon a project i then j should be scheduled at time period $itp(j)$ where

$$itp(j) \geq itp(i) + d(i) + l(j) \quad (6)$$

for $itp(i) > 0$, $d(i)$ is the duration of project i , and $l(j)$ is an integer called the *dependence time* that indicates how long before or after i finishes project j should be considered for scheduling. For example, if project i finishes at the time period 1 and $l(j) = 3$ then project j should be scheduled at the time period 5 or later, since a project ending time is equal to $itp(i) + d(i)$. The dependence time corresponds to the dependency months in the input instance.

We further define a set of restrictions that assures that project j will not be scheduled before $itp(i) + d(i) + l(i)$:

$$\sum_{t=1}^{itp(i)+d(i)+l(i)-1} x_{i,t} = 0 \quad \text{for } j \text{ dependent upon } i$$

We also assume that the time period $itp(i) + d(i) + l(i)$ is inside the PH boundaries.

Limited resources per year Each project demands a certain amount of resources, i.e. it has a certain cost, for each time period within its duration. The cost of a project i at period t of its activity is denoted by c_{it} . We denote by $C(t)$ the total cost demanded by all active projects during time period t . We can describe $C(t)$ by the equation:

$$C(t) = \sum_{k=1}^t \sum_{i=1}^I c_{i(t-k+1)} x_{ik} \quad (7)$$

For management purposes, the PH available resources are divided into sequences of time periods grouped by year. Let W_m denote the available amount of resources in year m , for $m = 1, \dots, T/12$. The sum of the costs for all projects at all time periods of an year must not be greater than the available resources for that year. We express this constraint using the inequality:

$$\sum_{j=12(m-1)+1}^{12m} C(j) \leq W_m, \quad \text{for } m = 1, \dots, T/12. \quad (8)$$

Notice that there are no constraints that restrict project costs at the time period $T+1$ or later. For this reason, projects whose cost cannot be supplied by the available resources along any usual time periods may end up being scheduled outside of the PH.

Resource categories Projects can also be classified into disjoint resource categories. The total amount of costs of projects belonging to category q should not surpass the available resources for that category, denoted by R_q .

$$\sum_{k=1}^T c_{ik}x_{ik} \leq R_q, \quad \text{for } i \in q \quad (9)$$

4 A Heuristics Solution

We present here a heuristics for solving the PPS problem. It was devised as a modification of the GRASP metaheuristic.

4.1 The GRASP Metaheuristic

The Greedy Randomized Adaptive Search Procedures [38, 19] (GRASP) is a multi-start metaheuristic that generates good quality solutions for many combinatorial optimization problems. The algorithm repeats two phases:

1. a construction phase in which a feasible solution is constructed.
2. a search phase in which the algorithm finds a locally optimal solution by iteratively checking whether there is a better solution in the neighborhood of the previously constructed solution.

Several initial feasible solutions are constructed and improved upon in order to reach locally optimal solutions. The best solution is then chosen.

During the construction phase a solution is constructed from an empty solution by iteratively choosing a solution element from a list of candidate elements. The list of candidates is formed by all elements capable of being included in the present solution, sorted by the value of a greedy function that measures the benefit of including each element in the solution at the present stage. At each iteration of the construction phase, the list of candidate elements is reordered by recalculating the element benefits and then an element is randomly chosen from a first segment of the list and is then included into the solution. By randomly choosing an element, the construction procedure avoids finding identical solutions at each repetition of the construction phase. When it terminates, the iterative procedure has generated a solution that is finally checked for its feasibility. If not feasible, the solution may be repaired and the algorithm can proceed to the search phase, using the constructed solution as the initial point in a local search.

The search phase starts by taking the solution found in the construction phase and calculating the set of solutions in a neighborhood of that solution. Then the algorithm iteratively selects a better solution in terms of the objective function in

that neighborhood, makes it the new current solution, and finds the set of solutions in the neighborhood of the new current solution. The cycle repeats until there is no better solution to be found in the neighborhood of the current solution.

The number of repetitions of the construction and search phases is defined as a parameter for the GRASP algorithm. At the end of each repetition of the construction and search phases, the algorithm verifies whether the currently found locally optimal solution is better than the best overall solution and replaces it accordingly.

Despite its apparent simplicity, GRASP seems to perform empirically well both in terms of solution quality and running-time, when compared to other simple search algorithms that search for local optima starting from a randomly generated solution. One of the reasons for the algorithm's good solutions quality and performance may stem from the greedy randomized strategy used to construct the initial solutions that feed the local search. Such initial solutions tend to be near local optimal solutions. Moreover, GRASP's random procedure for constructing an initial solution allows a good coverage of the solution space, so that the construction phase can be viewed as a sampling technique applied to the solution space. The better the sample of constructed solutions, the higher the chance of finding a solution nearer to one of the best solutions. GRASP has also the advantage of being easily coded, as it involves only a few parameters to be set and can be implemented without much effort.

Several techniques have been proposed to enhance different aspects of the basic GRASP metaheuristics [19]. Among these techniques, *cost perturbation* [19], *reactive GRASP* [36, 37], *path-relinking* [22], and hybridization with other heuristics [20] have provided good results.

4.2 A GRASP variant for PPS: kCRGH

We propose a simple variant of the GRASP heuristic for solving the PPS problem, here called the k -random cascading risk greedy heuristics (kCRGH). Instead of executing two distinct phases of construction and local search, the algorithm kCRGH implements a slightly modified version of the construction phase, distinct from the basic one in two aspects:

1. the construction procedure is performed in a way that guarantees that the solution found is feasible, without recurring to a repairing procedure at the end of the construction phase;
2. there is no update of element benefits and reordering of the list of candidate elements; and
3. there is no local search for better solutions in a neighborhood of a constructed solution.

The heuristics is based on the idea of sorting all possible pairs of projects and time periods within the PH. The parameter that measures the contribution of scheduling a project at a certain time period depends not only on a project controlled risk, but also on the project total cost, the project dependence to other projects, and whether the project controlled risk takes effect inside the PH.

When a portfolio contains interdependent projects, a single project controlled risk value does not reflect the potential total amount of controlled risk by chains of dependent projects. In other terms, if a project j has a greater controlled risk value than and is dependent upon a project i , the potential controlled risk not gained by **not** scheduling i at a certain time period should be measured considering not only R_i , the risk controlled by i , but also R_j , the risk controlled by project j , since j will be scheduled only if i is also scheduled. We define the *cascading risk* of a project i , denoted by Q_i , as

$$Q_i = R_i + \sum_{j \text{ depends upon } i} Q_j. \quad (10)$$

The sum of cascading risks of dependent projects contributes to the cascading risk of a project. The cascading risk better reflects the relevance of choosing a project, since it does not consider it in isolation, but takes into account the controlled risks of its dependent projects. Moreover, finding cascading risks does not pose a significant impact on the optimization performance, since they can be easily computed *a priori*.

The controlled risk of a project is not informative enough to decide whether it should be scheduled earlier than another project without comparing each project's cost. A project with a large value of controlled risk might not be more cost-effective in terms of controlled risk by unit of cost than another project. The ratio of the number of time periods in which a project controlled risk takes effect with respect to the duration of the PH is also relevant to the scheduling decision since the earlier a project is scheduled at, the longer a project contributes by controlling a certain amount of risk in the PH. To assess the cost-effectiveness and the duration of the effective risk controlled by scheduling a particular project, we define the *benefit* of a project i , scheduled to start at the time period t , denoted by $b(i, t)$, as the function:

$$b(i, t) = \frac{T - t - d(i)}{T} \times \frac{Q_i}{1 + \sum_{k=1}^T c_{ik}} \quad \text{for } 1 \leq t \leq T + 1.$$

The function $b(i, t)$ expresses the amount of cascading risk gained by a unit of cost weighted by the ratio of time periods when the controlled risk is effectively used over the total duration of T time periods. Since it is possible that a project has no costs, we add 1 to a project total costs to avoid a division by zero. We allow t to assume the value $T + 1$ in the benefit function, as a practical convention for allowing the assignment of a time period value outside of the PH to a project, that is, to denote a project not included in the portfolio.

kCRGH is a greedy algorithm that, firstly, pre-process the input data to schedule the mandatory projects, and then constructs a portfolio by iteratively choosing a pair (i, t) at random among the first k pairs from a list of candidate elements of the portfolio solution, that is, project and starting time pairs sorted by their benefit value and such that the inclusion of the pair in the portfolio keeps the solution feasible. The heuristics takes into consideration the interdependence among projects and the cascading risk in order to construct the list of sorted pairs (SP). Instead of building a list of pairs (i, t) sorted by project controlled risks, the list is sorted based on the benefit of each pair (i, t) .

Two properties of kCRGH are worth emphasizing:

- Randomness: by choosing a pair randomly, we expect a smaller chance of being trapped in a local optimum solution.
- Usage of cascading risks: by adopting cascading risks instead of single project controlled risks, the heuristics avoid situations in which the scheduling of a low risk project, to which a chain of high risk projects depends, is postponed and scheduled later than projects with slightly higher controlled risks. The resulting solution found by ignoring project dependencies would have a lower objective function value than a solution generated when considering projects interdependencies.

The pseudo-code for the kCRGH routine is presented as Algorithm 1.

Algorithm kCRGH termination follows trivially from the fact that one pair is removed from the SP list at each *while* iteration and that the original SP list never gains new pairs. Despite the removal of pairs, the relative order between two pairs in the SP is never altered. The algorithm guarantees that, eventually, any project will be scheduled even if outside the PH, that is time period $T + 1$, because project's costs at $T + 1$ are not restricted by any constraint. For this same reason, i.e. that the last time period resources cover any sum of costs, the algorithm always finds a feasible solution. The running time of the algorithm is bounded by $\Theta(I \times T \log I \times T)$.

The list SP is built in such a way that the following property holds: for pairs $SP[h] = (i, t)$ and $SP[l] = (j, r)$, $h < l$ if and only if $b(i, t) \geq b(j, r)$ or $t < r$. In other terms, lower indexed pairs in the SP have greater benefit values at earlier time periods than higher indexed pairs. The ordering in SP provides the greedy aspect of the algorithm by favoring the scheduling of the most beneficial project at its earliest time period.

When a mandatory project is included in the portfolio, the kCRGH algorithm does not check whether the available resources are enough to cover its cost. We do not include a resource availability verification since this feature is not implemented at the current stage of the prototype software that implements the kCRGH heuristics. At the present implementation, the prototype emits a warning message and proceeds its

Algorithm 1 Obtain a solution for the PPS problem

1. Initialize an empty portfolio P .
 2. **for all** mandatory projects **do**
 3. Schedule the mandatory project to its initial scheduled time.
 4. Remove from the available resources mandatory project's all required resources by time period.
 5. **end for**
 6. Calculate the cascading risk Q_i of each project i .
 7. Construct the SortedPairs (SP) list of $I \times (T + 1)$ project and starting time pairs, sorted according to each pair benefit value and starting time.
 8. **while** SP is not empty **do**
 9. Get a pair (i, t) randomly chosen from the first k pairs at the top of the SP list.
 10. Remove the pair (i, t) from the SP.
 11. **if** i was already scheduled **then**
 12. Discard pair (i, t) and go to the next iteration.
 13. **end if**
 14. **if** adding (i, t) keeps the solution feasible **then**
 15. Make $P = P \cup \{(i, t)\}$ and discard (i, t) .
 16. **else**
 17. Discard pair (i, t) .
 18. **end if**
 19. **end while**
 20. **return** Portfolio P .
-

execution when the mandatory project costs surpass the available resources, allowing all mandatory projects to be included in the portfolio.

As discussed before, the kCRGH heuristics is distinct from the basic GRASP heuristics by constructing a single solution only, instead of repeatedly searching for several locally optimal solutions. The inclusion of a search phase and the iterative repetition of both construction and search phases to fully implement the GRASP metaheuristic could provide better solutions, even though it might increase the running time of the algorithm.

5 Experiments

We performed experiments to evaluate the objective function value and the performance of kCRGH when solving the PPS problem. We assume that the heuristic running-time and the objective function value are good indicators of its quality.

For assessing the indicator values, we devised an experiment that consisted of comparing the initial portfolio solution for a real-world instance and the solution generated by kCRGH for the same instance. The initial portfolio is the one found in the input instance as defined in the spreadsheet provided by AES-Tietê. More specifically, we compare the objective function values for the initial portfolio solution and the solution found by the kCRGH. We also present the kCRGH running time for reference only, since there is no proper estimate for the time it takes for finding the initial portfolio.

For a better assessment of the indicators, we need a sample of real world portfolios to perform the comparisons. We used the real world instance provided by AES-Tietê as a seed for generating similar “real world”-like instances by a procedure of disturbing the seed portfolio data. In order to comply with AES-Tietê confidentiality policies, we performed a simple data obfuscation procedure on the costs and available resources in the seed portfolio. The real world seed portfolio, after the obfuscation procedure, is characterized by the following parameters:

- Portfolio cumulative total risk: $2.34M$.
- Number of projects uploaded: 1573.
- Planning horizon: years of 2012, 2013, 2014, 2015, and 2016.
- Number of mandatory projects: 1354.
- Number of dependence relations: 0.
- Projects with non-zero lead time: 0.
- Average cost of each project by year (BRL): $819.98M$, $1144.63M$, $1058.28M$, $828.0M$, and $902.98M$.

- Total sum of costs for all years (BRL): around 5741.89 million.

The data disturbing procedure consisted of applying the following modifications to the input instance data:

Project cost an increase or decrease by a 10% factor with probability of 50%; otherwise no modification at all is applied to the project cost value.

Project dependency a 10% of chance of changing the project dependency to an identifier from the list of projects identifiers, including 0, i.e. no dependence. We exclude from the list the project itself and other projects previously involved in dependencies, in order to avoid cyclic dependencies.

Total risk an increase or decrease by a 10% factor of a project total risk.

Mandatory projects a 10% of chance of changing a project mandatory status.

Using these modifications, we produced 20 portfolio instances that were submitted to the kCRGH algorithm. We adopted the following parameters for the optimization:

- CAPEX available resources (BRL): 2657M, 2524M, 2391M, 2258M, and 2125M. The CAPEX available resources were obtained by taking the total cost for all CAPEX projects in all years in the seed portfolio and making it the first year amount of available resources. The subsequent yearly resources were obtained by decreasing the first year value by 5% per year.
- OPEX available resources (BRL): 648.44M, 616.02M, 551.17M, 549.24M, and 518.75M. The OPEX resources were obtained using the same procedure adopted for CAPEX resources, but applying it to OPEX projects.
- The parameter k in the kCRGH is 5.
- The number of years in the PH is 5, in all input instances.

After applying the heuristics to each portfolio instance, we identified the instance whose heuristics solution had the largest gain in terms of objective function value when compared to the initial portfolio. The characteristics of the portfolio with the largest gain were:

- Portfolio cumulative total risk: 2.39M.
- Number of projects uploaded: 1573.
- Planning horizon: 2012, 2013, 2014, 2015, and 2016.
- Mandatory projects: 164.

Solving Method	Objective function value	Running time
Empirical	2,396,403	–
kCRGH	3,225,826	31.3 s

Table 3: Objective function value and running time for obtaining an empirical and a kCRGH solution for a real-world instance with a PH of five years and 1573 projects.

- Dependence relations: 0.
- Projects with lead time: 0.
- Average cost of each project by year (BRL): 815.97M, 728.28M, 678.75M, 526.80M, and 900.31M.
- Total sum of costs for all years (BRL): around 857.8 million.

Table 3 shows the objective function value and the kCRGH running time.

The objective function value for the solution found by the kCRGH is around 35% greater than the value for the initial portfolio. Figure 3 shows in the same chart a comparison between the cumulative total risk curves for the initial portfolio, indicated by a dashed curve, and the optimized portfolio produced by the kCRGH, represented in the graph by a solid curve. In the optimized portfolio, a larger amount of total risk is covered at an earlier stage in the PH, while in the initial portfolio, the amount of project total risks starts to increase at the last years of the PH. When the initial portfolio cumulative risk reaches the same level of the optimized portfolio at the end of the PH, the total amount of controlled risk over the time for the optimized portfolio is superior than the amount of controlled risk for the initial portfolio.

To better understand some of the properties of the optimized portfolio and the initial portfolio we refer to their cost and total risk curves. Figure 4 shows the total risk curves of the initial portfolio, dashed, and the portfolio produced by the heuristics depicted by the solid curve. Observe that in the first half of the PH the optimized portfolio schedules projects with a greater amount of controlled risk than the initial portfolio, while in the second half of the PH, both portfolios present a reduction in the amount of controlled risk and the initial portfolio has a greater amount of total risk controlled when compared to the optimized portfolio. Such properties reflect the heuristics strategy of scheduling projects that control a large amount of total risk as earlier as possible in the PH.

Figure 5 shows the cost curves for the initial (dashed curve) and optimized (solid curve) portfolios. While in the initial portfolio the total amount of required costs at each year is about the same, in the optimized portfolio the largest share of costs are required at the end of the second year of the PH. Again, the concentration of costs at the first two years of the PH is a consequence of the heuristics greedy strategy.

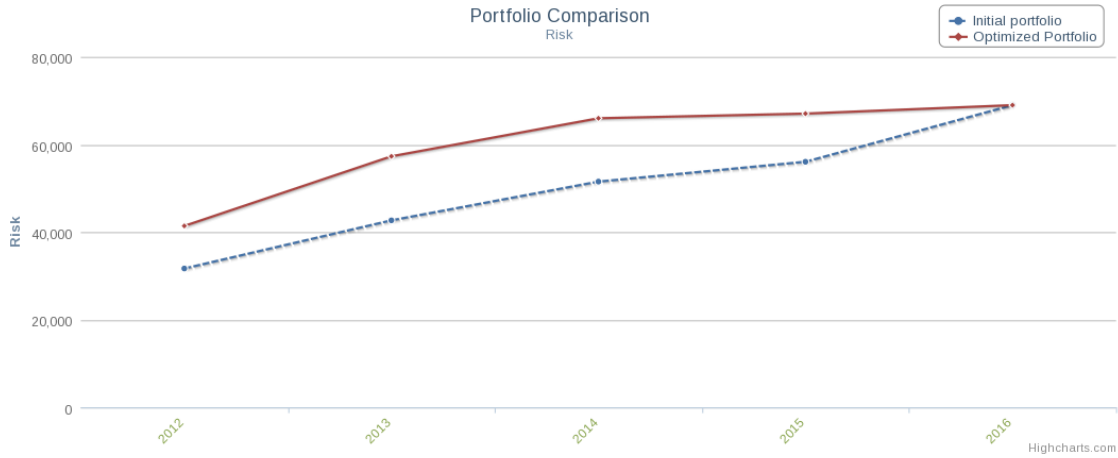


Figure 3: Comparison between the cumulative total risk curves of the initial portfolio found in the input instance depicted in the dashed curve, and the optimized portfolio produced by the kCRGH shown in the solid curve.

6 A Decision Support System Prototype

We developed a prototype for a decision support system (DSS), called the *AES Project Manager*, as a part of a set of activities involved in an ongoing research project whose aim to improve electricity generation industry management practices.

A DSS oriented to solve the PPS problem generally involves a set of functionalities that cover the series of phases found in the portfolio selection process. Archer and Ghasemzadeh [4] identified three phases in the portfolio selection process:

1. *strategic considerations* – considerations involving external and internal factors to a company, including the marketplace and the company’s strengths and weaknesses.
2. *individual project evaluation* – methods used in the measurement of the individual project contribution to a portfolio objective function.
3. *portfolio selection* – comparison and selection of portfolios according to some criteria that establishes a ranking among them.

The design of the prototype focused on the individual project evaluation and portfolio selection phases of the portfolio selection process. A more extensive decision support system for the PPS problem would involve an analysis of the complete portfolio selection process and probably an adaptation of an existing framework, such as the one proposed by Archer and Ghasemzadeh [3] or Chu et al. [12]. An example

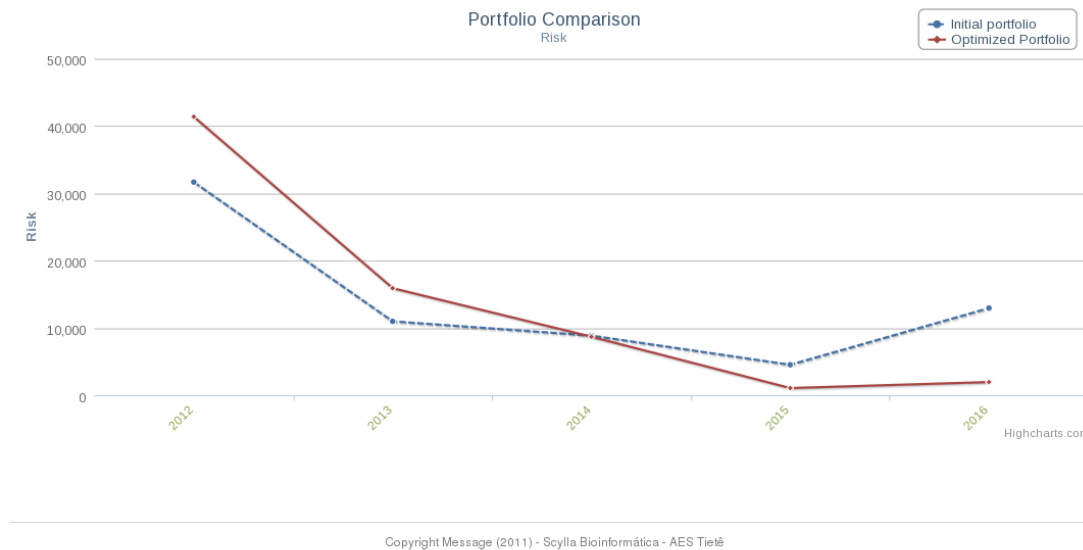


Figure 4: Comparison between total risk curves of the initial portfolio depicted in the dashed curve, and the optimized portfolio produced by the heuristics shown in the solid curve.

of a functionality not implemented in the present prototype is a *checking system* that would validate project parameters according to user defined rules and check for inconsistencies. Rather, the prototype implements a simple data validation that takes place while importing a list of projects.

The AES Project Manager prototype implements the kCRGH heuristics, besides portfolio data management and visualization. The software is a client-server, dynamic web application developed in **Python** using the *Django 2.4* framework in accordance with the *MVC* architecture. Its persistence is based in the integration of *Django* and *PostgreSQL*. The presentation layer makes use of the well established *JQuery* framework and a chart generation javascript framework called *Highcharts JS*.

The prototype was designed to offer the following features:

1. Portfolio and Project management – importing and exporting of portfolios, project and portfolio CRUD (create, retrieve, update, and delete) operations, and project search mechanisms.
2. Portfolio optimization – definition of input instance parameters and execution of the kCRGH heuristics.
3. Portfolio comparison – presentation of charts that facilitate comparing aspects of two or more portfolios.

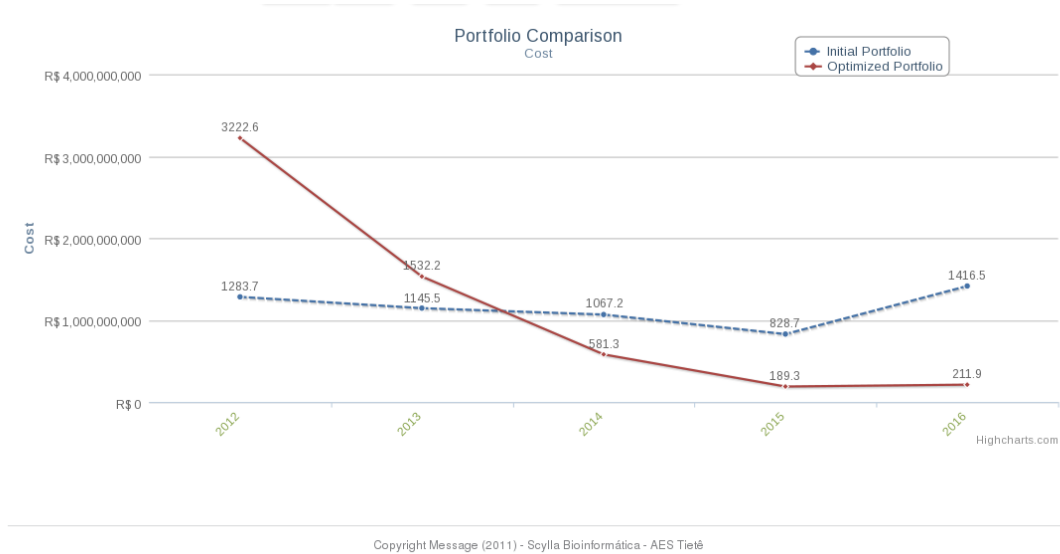


Figure 5: Yearly cost curves of the initial portfolio depicted in the dashed curve, and the optimized portfolio produced by the heuristics shown in the solid curve.

4. Data visualization – charts and tables presenting important values, such as, monthly and yearly total costs, total risks, and cumulative total risks.

Figure 6 shows the AES Project Manager prototype main page interface.

Portfolio and project management The portfolio and project management functionalities include features such as the importing of the spreadsheet data describing the PPS problem input instance, the exporting of a portfolio to a spreadsheet file, and the CRUD operations over projects and portfolios. Figure 7 shows a screenshot of the list of projects interface.

The portfolio and project management functionalities facilitate the evaluation of a recurring scenario in the portfolio selection process, when the decision maker wants to test the effects of a few alterations on some project parameters, e.g. a project risk or cost. After introducing the modifications, the decision maker can run the optimization algorithms and compare the results.

Portfolio optimization The portfolio optimization functionality is responsible for the definition of the PPS problem input instance parameters and the execution of the kCRGH heuristics. When a user selects the optimization feature for the current set of projects, the system shows a form where the user can define the main parameters for the PPS input instance, such as the amount of resources available for each year



Figure 6: AES Project Manager prototype main page interface.

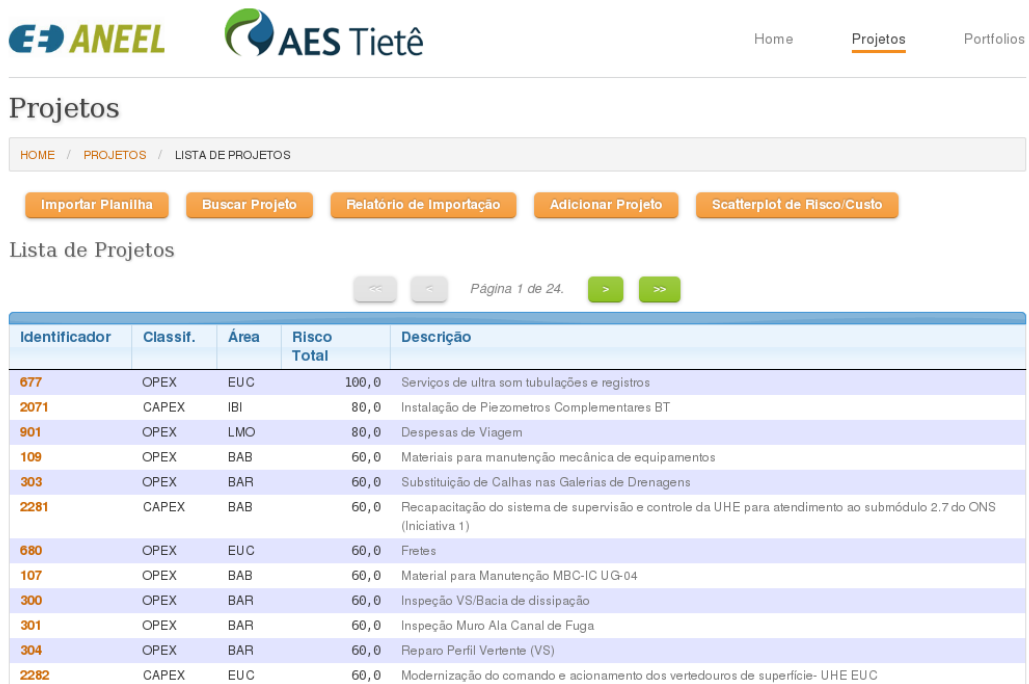


Figure 7: Prototype interface showing a list of projects.

Otimização

Nome do Portfolio*:

Descrição do Portfolio*:

Número de anos considerados:

Número de projetos na escolha aleatória:

Limites máximos de Custos:

	2012	2013	2014	2015	2016
CAPEX	<input type="text" value="457,926,056"/>	<input type="text" value="435,029,753"/>	<input type="text" value="412,133,450"/>	<input type="text" value="389,237,147"/>	<input type="text" value="366,340,844"/>
OPEX	<input type="text" value="1,383,163,881"/>	<input type="text" value="1,314,005,686"/>	<input type="text" value="1,244,847,492"/>	<input type="text" value="1,175,689,298"/>	<input type="text" value="1,106,531,104"/>
TOTAL	<input type="text" value="1,841,089,937"/>	<input type="text" value="1,749,035,439"/>	<input type="text" value="1,656,980,942"/>	<input type="text" value="1,564,926,445"/>	<input type="text" value="1,472,871,948"/>

Percentual do 1o ano: %

Figure 8: Screen containing a form for the definition of the optimization parameters. For instance, the user can define the amount of available resources at each year in the PH.

of the PH, as well as the name to be assigned to the new portfolio solution that will be found by the heuristics. The parameters that comprise the optimization form are:

1. portfolio name;
2. portfolio description;
3. number of years to be considered by the optimization algorithm;
4. number of projects used as the parameter k in the kCRGH;
5. available resources for CAPEX and OPEX projects at each year of the PH.

Some slots of the available resources input form can be filled automatically by simply choosing a functionality that applies a fixed percentage reduction to the first year resource values and attributes the reduced values to each subsequent year of the PH. Figure 8 depicts the optimization form.

The result of the execution of the optimization algorithm is a new portfolio solution for the specified PPS input instance defined by the user in the optimization form.

The main parameter that measures the benefit of the new portfolio solution is its cumulative total risk. Figure 9 shows the cumulative total risk chart of a portfolio.

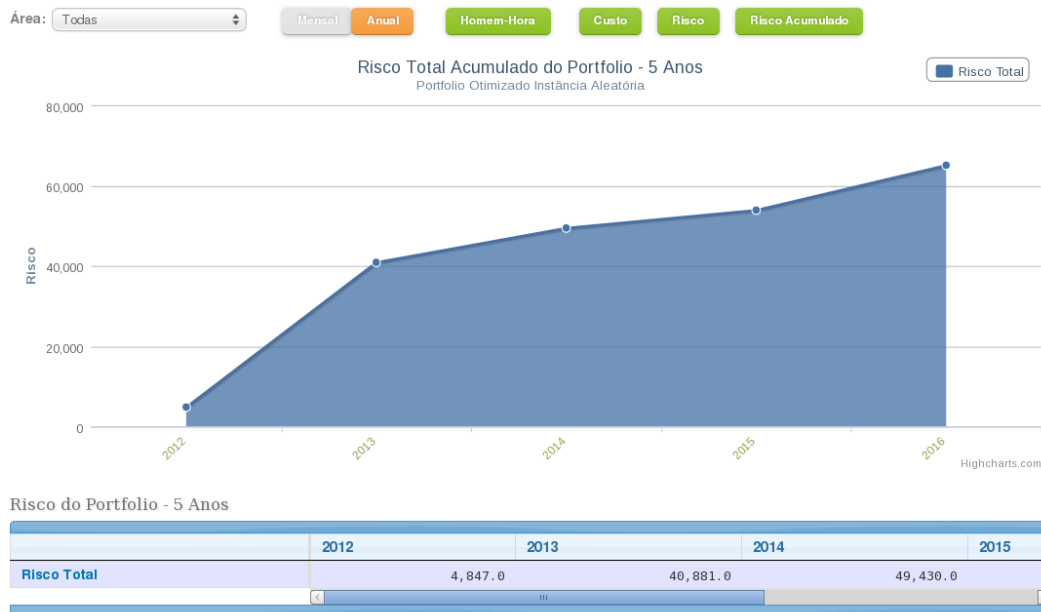


Figure 9: A screenshot of the cumulative total risk chart for a portfolio obtained by the optimization functionality.

Figure 10 shows the portfolio panel interface. It has two sections: the left side section shows a schedule of projects from the same project area chosen by the user, and the right side section shows the total cost and cumulative total risk charts for the portfolio. The schedule of projects is a grid that depicts each project in a row and each time period in the PH. When choosing a project area in the selection control found at upper left region on the screen, the user informs the software to filter the list of projects, so that the schedule exhibits the projects in the selected project area only. A bullet pin mark in a grid entry indicates that the corresponding project starts at the time period of the corresponding column. The portfolio panel allows a user to change a project schedule time by redefining the project start time period. Changes in cost and cumulative risk values in the portfolio are immediately shown in the right side charts as the user redefines a project schedule in the PH. This schedule edition functionality might help a decision maker to check the impact on the portfolio parameters when he makes slight changes in the present portfolio. This allows for the testing of different planning scenarios.

Portfolio comparison A fast and reliable way for comparing and presenting two or more portfolios is an essential feature in a decision support system for the PPS problem. To aid the decision maker, the presentation of comparisons is visually contrasting to the point that a simple chart can clearly distinguish two different

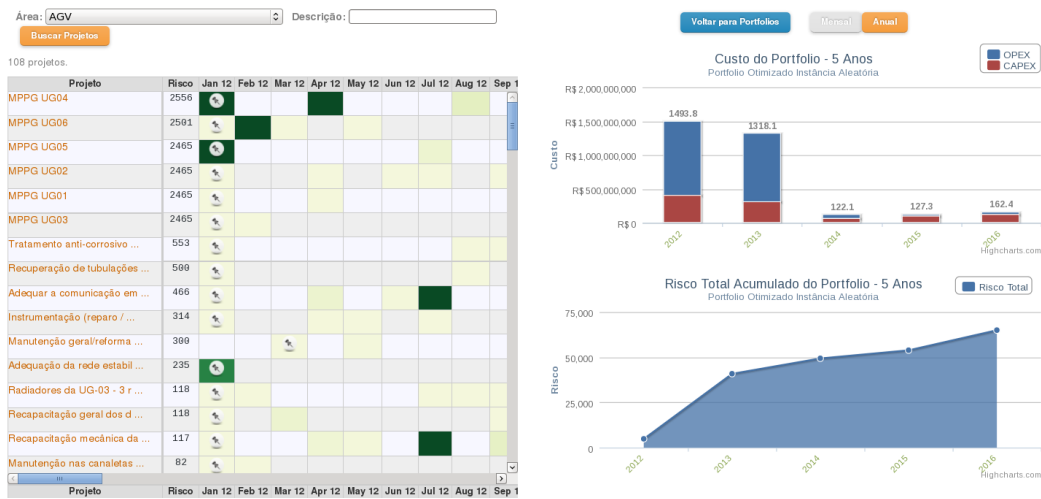


Figure 10: A screenshot of the portfolio panel interface. The left side of the screen depicts a grid representing the portfolio schedule, while the right side of the screen contains a total cost and cumulative total risk charts.

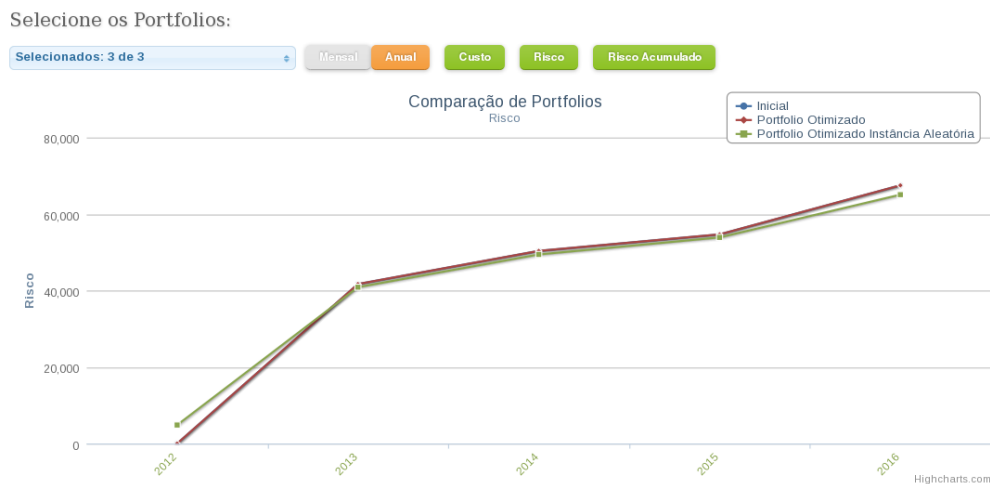


Figure 11: A screenshot of the chart showing the comparison between cumulative total risks from two distinct portfolios.

portfolios. Figure 11 shows a chart containing the cumulative total risk curves for two portfolios.

Data visualization Portfolio and project parameters relevant to the decision maker, including monthly costs, yearly costs, yearly total risks, and cumulative total risk,

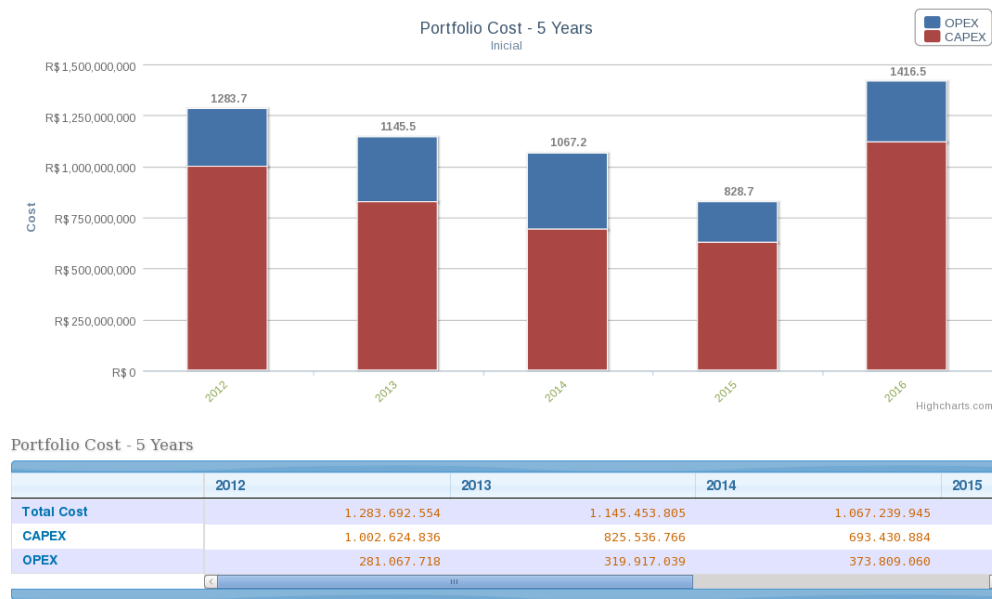


Figure 12: AES Project Manager prototype yearly total cost chart in the portfolio details interface.

are presented by charts and tables that can be invoked from the prototype using the *portfolio details* interface.

The portfolio details interface exhibits several informative aspects of a portfolio. The main components in the portfolio details interface is a chart and a corresponding data table, both containing the portfolio data relative to a certain parameter. At the top of the window there are control elements the user can set to modify which parameter should be exhibited in the charts and tables. The first control element is a selection box that allows one to filter the list of projects considered in the portfolio according to the selected project area. There is also a toggle button that defines whether the chart and table should show monthly or yearly period divisions. The *cost*, *risk*, and *cumulative risk* buttons define which of the respective charts and tables the interface should exhibit. Figure 12 shows the portfolio details window interface containing a chart and a table, both describing a portfolio yearly total cost. The amount of costs by OPEX projects are indicated in blue in the chart. CAPEX project costs are depicted in red in the column graph.

Figure 13 shows the same portfolio details interface, but now the interface depicts the parameter cost in terms of monthly time periods, both in the chart and table, for the first two years in the PH.

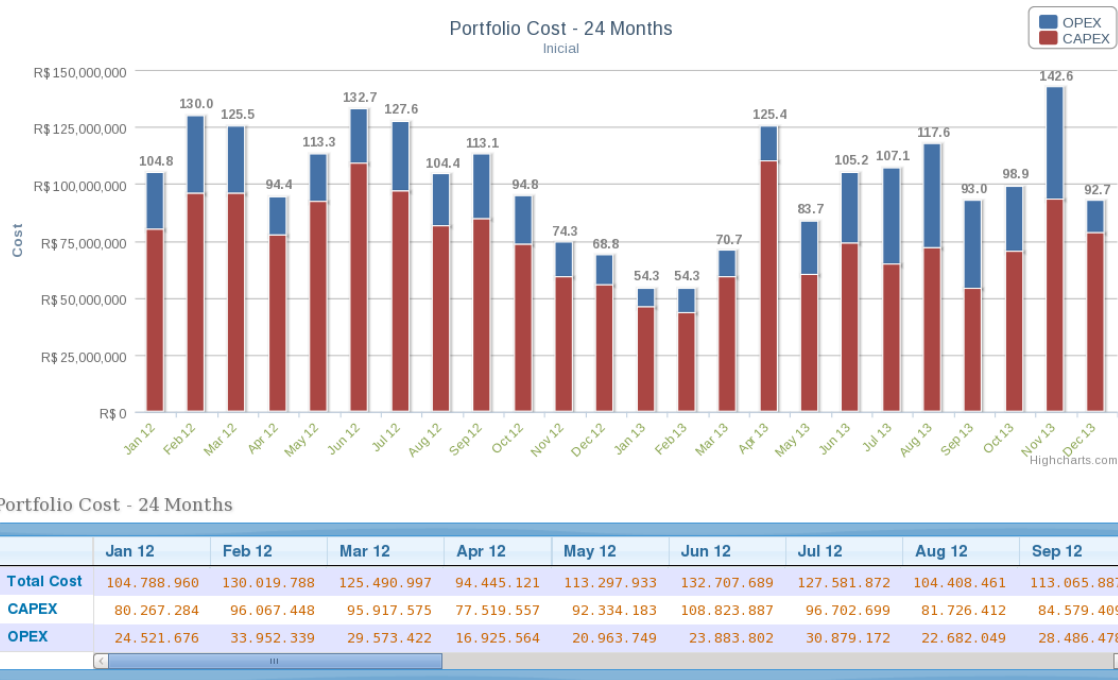


Figure 13: AES Project Manager prototype monthly total cost chart interface.

7 Conclusions

Project portfolio selection is a problem found in several relevant industries in a variety of different circumstances. We treated a particular PPS problem stemming from the electricity generation industry, and propose a heuristics, the kCRGH, based on the GRASP meta-heuristics for solving it. We performed experiments that confirmed that solutions produced by the heuristics are better than manually proposed solutions.

We also presented a DSS prototype that implements the heuristics and several features, including data visualization (tables and charts), data management, and total risk/cost charts. The DSS software prototype includes functionalities such as importing and exporting spreadsheets containing PPS instances and solutions, the search and construction a portfolio solutions using the proposed kCRGH heuristics, and portfolio edition allowing decision makers to apply modifications to a existing portfolio and recompute new solutions.

Using benefit feedback and adjusting the list of candidate elements in the kCRGH, may contribute to improve the heuristics quality and performance. Future work may also involve checking the heuristics stability in experiments involving randomly generated input instances.

8 Acknowledgment

The authors would like to thank the financial support of ANEEL (grant PD 0064-1021/2010).

References

- [1] A.T. Almeida and M. Duarte. A multi-criteria decision model for selecting project portfolio with consideration being given to a new concept for synergies. *Pesquisa Operacional*, 31(2):301–318, 2011.
- [2] G. Anandalingam and C.E. Olsson. A multi-stage multi-attribute decision model for project selection. *European journal of operational research*, 43(3):271–283, 1989.
- [3] N.P. Archer and F. Ghasemzadeh. A decision support system for project portfolio selection. *International Journal of Technology Management*, 16(1):105–114, 1998.
- [4] N.P. Archer and F. Ghasemzadeh. An integrated framework for project portfolio selection. *International Journal of Project Management*, 17(4):207–216, 1999.
- [5] N. Baker and J. Freeland. Recent advances in r&d benefit measurement and project selection methods. *Management Science*, pages 1164–1175, 1975.
- [6] N.R. Baker. R&d project selection models: an assessment. *IEEE Transactions on Engineering Management*, 21(4):165–171, 1974.
- [7] A.F. Carazo, T. Gómez, J. Molina, A.G. Hernández-Díaz, F.M. Guerrero, and R. Caballero. Solving a comprehensive model for multiobjective project portfolio selection. *Computers & operations research*, 37(4):630–639, 2010.
- [8] C. Carlsson, R. Fullér, M. Heikkilä, and P. Majlender. A fuzzy approach to r&d project portfolio selection. *International Journal of Approximate Reasoning*, 44(2):93–105, 2007.
- [9] A. Charnes and W.W. Cooper. Goal programming and multiple objective optimizations: Part 1. *European Journal of Operational Research*, 1(1):39–54, 1977.
- [10] A. Charnes, W.W. Cooper, and R.O. Ferguson. Optimal estimation of executive compensation by linear programming. *Management science*, 1(2):138–151, 1955.
- [11] A. Charnes, W.W. Cooper, and E. Rhodes. Measuring the efficiency of decision making units. *European journal of operational research*, 2(6):429–444, 1978.

- [12] P.Y.V. Chu, Y.L. Hsu, and M. Fehling. A decision support system for project portfolio selection. *Computers in industry*, 32(2):141–149, 1996.
- [13] R.G. Cooper. *Winning at new products: Accelerating the process from idea to launch*. Basic Books, 2001.
- [14] Y. Crama and M. Schyns. Simulated annealing for complex portfolio selection problems. *European Journal of operational research*, 150(3):546–571, 2003.
- [15] M.W. Dickinson, A.C. Thornton, and S. Graves. Technology portfolio management: optimizing interdependent projects over multiple time periods. *Engineering Management, IEEE Transactions on*, 48(4):518–527, 2001.
- [16] K.F. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, and C. Stummer. Pareto ant colony optimization with ilp preprocessing in multiobjective project portfolio selection. *European Journal of Operational Research*, 171(3):830–841, 2006.
- [17] B.P.M. Duarte and A. Reis. Developing a projects evaluation system based on multiple attribute value theory. *Computers & operations research*, 33(5):1488–1504, 2006.
- [18] A. Fernández Carazo, T. Gómez Núñez, F.M. Guerrero Casas, and R. Caballero Fernández. Evaluación y clasificación de las técnicas utilizadas por las organizaciones, en las últimas décadas, para seleccionar proyectos. *Revista de métodos cuantitativos para la economía y la empresa*, 5(1):67–115, 2008.
- [19] P. Festa and M.G.C. Resende. Grasp: basic components and enhancements. *Telecommunication Systems*, 46(3):253–271, 2011.
- [20] Paola Festa and Mauricio G. C. Resende. Hybrid grasp heuristics. In *Foundations of Computational Intelligence (3)*, pages 75–100. 2009.
- [21] G.E. Fox, N.R. Baker, and J.L. Bryant. Economic models for r and d project selection in the presence of project interactions. *Management science*, pages 890–902, 1984.
- [22] F. Glover. Tabu search and adaptive memory programming—advances, applications and challenges. *Interfaces in computer science and operations research*, 1, 1996.
- [23] J. Gustafsson and A. Salo. Contingent portfolio programming for the management of risky projects. *Operations research*, pages 946–956, 2005.
- [24] W.J. Gutjahr, S. Katzensteiner, P. Reiter, C. Stummer, and M. Denk. Multi-objective decision analysis for competence-oriented project portfolio selection. *European Journal of Operational Research*, 205(3):670–679, 2010.

- [25] M. Hartmann and A. Hassan. Application of real options analysis for pharmaceutical r&d project valuation—empirical results from a survey. *Research Policy*, 35(3):343–354, 2006.
- [26] K. Heidenberger. Dynamic project selection and funding under risk: A decision tree based milp approach. *European Journal of Operational Research*, 95(2):284–298, 1996.
- [27] A.D. Henriksen and A.J. Traynor. A practical r&d project-selection scoring tool. *Engineering Management, IEEE Transactions on*, 46(2):158–170, 1999.
- [28] T.Y. Hsieh and H.L. Liu. Genetic algorithm for optimization of infrastructure investment under time-resource constraints. *Computer-Aided Civil and Infrastructure Engineering*, 19(3):203–212, 2004.
- [29] G. Hu, L. Wang, S. Fetch, and B. Bidanda. A multi-objective model for project portfolio selection to implement lean and six sigma concepts. *International journal of production research*, 46(23):6611–6625, 2008.
- [30] B. Jackson. Decision methods for selecting a portfolio of r&d projects. *Research Management*, 26(5):21–26, 1983.
- [31] R.T. Marler and J.S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.
- [32] J.P. Martino. *Research and development project selection*, volume 23. John Wiley & Sons, 1995.
- [33] R.G. Mathieu and J.E. Gibson. A methodology for large-scale r&d planning based on cluster analysis. *Engineering Management, IEEE Transactions on*, 40(3):283–292, 1993.
- [34] L.M. Meade and A. Presley. R&d project selection using the analytic network process. *Engineering Management, IEEE Transactions on*, 49(1):59–66, 2002.
- [35] R.P. Mohanty, R. Agarwal, A.K. Choudhury, and M.K. Tiwari. A fuzzy anp-based approach to r&d project selection: a case study. *International Journal of Production Research*, 43(24):5199–5216, 2005.
- [36] M. Prais and C.C. Ribeiro. Parameter variation in grasp implementations. In *Extended abstracts of the third metaheuristics international conference*, pages 375–380, 1999.
- [37] M. Prais and C.C. Ribeiro. Parameter variation in grasp procedures. *Investigación Operativa*, 9:1–20, 2000.

- [38] Mauricio G. C. Resende. Greedy randomized adaptive search procedures. In *Encyclopedia of Optimization*, pages 1460–1469. 2009.
- [39] R. Santhanam and J. Kyparisis. A multiple criteria decision model for information system project selection. *Computers & Operations Research*, 22(8):807–818, 1995.
- [40] C. Stummer and K. Heidenberger. Interactive r&d portfolio analysis with project interdependencies and time profiles of multiple objectives. *Engineering Management, IEEE Transactions on*, 50(2):175–183, 2003.
- [41] C.K. Suh, E.H. Suh, and K.C. Baek. Prioritizing telecommunications technologies for long-range r&d planning to the year 2006. *Engineering Management, IEEE Transactions on*, 41(3):264–275, 1994.