

INSTITUTO DE COMPUTAÇÃO

UNIVERSIDADE ESTADUAL DE CAMPINAS

Project Scheduling Optimization in Energy Generation Plants

C. Mira P. Viadanna M. A. Souza
J. Meidanis A. Moura G. A. C. Lima
R. P. Bossolan

Technical Report - IC-14-11 - Relatório Técnico

July - 2014 - Julho

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

Project Scheduling Optimization in Energy Generation Plants

Cleber Mira* Paulo Viadanna* Maria Angélica Souza* João Meidanis*
Arnaldo Moura† Gabriel Costa Lima‡ Renato Bossolan§

Abstract

The problem of choosing from a set of projects which ones should be executed and when they should start, depending on several restrictions involving project costs, risks, limited resources, dependencies among projects, and aiming at different, even conflicting, goals is known as the project portfolio selection (PPS) problem. We study a particular version of the PPS problem stemming from the operation of a real power generation company. It includes distinct categories of resources, intricate dependencies between projects, which are specially important for the management of power plants, and the prevention of risks. We present an algorithm based on the GRASP meta-heuristic for finding better results than a manual solution produced by specialists. The algorithm yielded solutions that decreased the risk by 47%, as measured by the company's standard methodology.

1 Introduction

Decision makers are usually confronted with the problem of choosing from a large set of projects, according to a combination of criteria and given a limited amount of resources. A project portfolio consists of a subset of the given projects, together with a schedule for their execution over a given period of time, and in such a way that all restrictions are obeyed. This is known as the *project portfolio selection* (PPS) problem. There are several formulations for the PPS problem [6] depending on specific constraints, objective functions, resource availability, labor policies, and timing assumptions. However, since these formalisms usually share characteristics, a single-objective function PPS problem can be summarized as follows: given the set of proposed projects, find a valid project portfolio that optimizes a given objective function over all other portfolios. With multiple objective functions present, one wants to find a number of portfolios that reasonably achieve partial goals, such as portfolios that rest in a Pareto boundary.

The single-objective function version of the PPS problem is a classical optimization problem [4] with applications in finances [5], research and development [11, 16, 19], software development [13], and project management [2, 3, 15], among many other fields. There is an extensive literature describing several approaches to model PPS problem variants [9, 1]. Although not as studied as the single-objective function version, the multi-objective version of the PPS problem

*Scylla Bioinformatics

†Institute of Computing, Unicamp

‡Mechanical Engineering Department, Unicamp

§AES-Tietê

has gained increasing interest in recent years [6, 7, 10, 20]. A list of methods that can be used for solving the single- and multi-objective versions of the PPS problem is also available [18, 17].

In this work we focus on describing a particular single-objective PPS problem variant stemming from the electrical power generation industry. This variant brings up some specific characteristics, such as mutual exclusion dependencies between projects and project grouping, which affects the objective function in different ways. The identification of these particular characteristics is a result of a careful and detailed elicitation of constraints by expert engineers. The set of all constraints is too daunting to be taken into consideration by decision makers who construct portfolios manually. Therefore, computational approaches to treat this problem are particularly important.

In Section 2 we outline the main characteristics of the particular PPS problem studied here. In Section 3 we present the problem input and describe the parameters needed to model and solve the problem. Section 4 describes the problem constraints. We discuss in Section 5 PPS problem optimization goals. Section 6 presents a heuristic to find good solutions for the variant of the PPS problem treated here. We show the results of experiments in Section 7, and summarize in Section 8.

Nomenclature

T	size of planning horizon (PH) in months
I	set of projects
$[n]$	interval $\{1, 2, \dots, n\}$
d_p	duration (in months) of project p
I_P	set of projects in portfolio P
m_p	starting month of project p in portfolio P
e_p	last month of execution of project p
l_p	lead time of project p
p_m	prescribed initial month for project p in the manual solution
G_w	group of projects controlling warning point w
W	set of all warning points
f_w	latest over all last months of execution of projects controlling w
D_w	deadline associated to high-risk warning point w
Q	set $\{CAPEX, OPEX\}$
$r_{p,m}$	resource consumption of project p in the m -th month of its duration
$P_{t,q}$	set of projects of class q active in month t
$c_{P,t,q}$	resource consumption of class q on month t for portfolio P
$C_{P,y,q}$	yearly version of $c_{P,t,q}$ above
$S_{y,q}$	resources available for class q and year y
$g(u, t)$	number of generating units halted at plant u on month t
$g_L(u, t)$	same as $g(u, t)$ but due to long-term projects
R_w	risk of warning point w
W_P	set of warning point effectively controlled by portfolio P
$R(P)$	area under the risk curve for portfolio P
V_I	set of viable portfolios for instance I

2 Problem Description

We describe a specific version of the PPS problem stemming from a real case occurring at an electrical power generation company, AES-Tietê, operating in Brazil.

2.1 Projects

A power generation company is periodically confronted with the selection and scheduling of projects that contribute to achieve diverse, usually conflicting, objectives such as:

1. Comply with government regulations;
2. Minimize risk exposures;
3. Increase return over investments;
4. Reduce operation costs;

There are two kinds of projects at AES-Tietê:

1. *Risk management projects*: these projects are responsible for guarding and mitigating risks that might arise during the power generation operation. They impact operational tasks, as well as other activities such as the refurbishing of installed equipment, the acquisition of new equipment, or the improvement of maintenance procedures. They can be scheduled with a certain amount of freedom, respecting a number of operational constraints.
2. *Non-risk management projects*: these are projects that do not contribute to the prevention or mitigation of risks. Examples are projects for the acquisition of office supplies, of personnel safety equipment, or of employee uniforms, among others. Non-risk management projects also include projects that help the company reach its strategic goals, *e.g.*, the construction of a new power plant, or projects that attend exclusively to regulatory demands, such as environmental legislation concerning reservoir area, production capacity, and power availability. Due to their nature, projects of this kind cannot be scheduled at will by the decision maker; they must follow a strict, predefined schedule.

The selected projects must be scheduled to start at specific months within a period of time called the *planning horizon* (PH). The extent of the planning horizon, denoted by T , is typically 60 months in our case study. Non-risk management projects are considered *mandatory*, that is, they must start at a prescribed initial month within the PH. Risk management projects are not mandatory, which means that their properties — such as costs, risk guarding potential, and interdependencies — are scrutinized by a decision maker, who decides if, and when, to schedule them for execution.

Projects are executed under the supervision of an hierarchy of administrative units. At the top of this hierarchy, we have *divisions*. Each division is formed by units called *areas*. An area can be a purely administrative unit, like an information and technology center or a general operations center. Alternatively, an area can be an operational unit, such as a power plant. Knowing which power plant will supervise a project's execution might impact its scheduling.

2.2 Warning points

The generation of electricity is a complex process. As such, several undesired events might occur *e.g.*, equipment failure, violation of government regulations, or accidents. In our case study, prevention against the occurrence of undesired events is a top priority and there is a consolidated process to achieve this control, consisting of the following steps:

1. Elicit *warning points*, that is, identify specific conditions that may cause undesired events. Warning points are then used to guide specialists when they elaborate measures to prevent the occurrence of undesired events.
2. Assign to each warning point an amount of risk according to a standard set of criteria. Such criteria might include the probability of occurrence of their associated undesired events, how severe a safety hazard it can be, the severeness of occasional environment damages that might follow, or threats to the company image, among others. The amount of risk should be proportional to the impact the occurrence of the corresponding undesired events may cause, if the warning point is not controlled. Warning points are classified into two risk categories, namely, *critical* and *non-critical*. The non-critical category includes warning points with either a tolerable or moderate risk. The critical category includes warning points with an intolerable risk. The risk category is assigned by expert personnel and will be seen to impact the scheduling of a project.
3. Devise projects that *control* the warning points, in order to prevent or mitigate the impact of the associated undesired events. Usually, a warning point is controlled by a *group of projects*. In our present modeling, all projects in a group are taken to contribute equally for controlling the risk associated to the warning point. According to the available data, and following field engineers advice, this was deemed reasonable in a first approach. Notice that a project may belong to more than one group of projects, and therefore it may contribute to control the risk of several warning points. The total risk controlled by a project is obtained by summing all the shares of risk the project controls in all groups it belongs to.

In this PPS problem variant we focus on controlling the risk assigned to warning points. The risk associated to a warning point can be considered completely averted only after all projects in its controlling group have been completely executed. Moreover, high-risk warning points must have their associated risk completely controlled by the end of a given *deadline*, which is a specific month along the PH. Only high-risk warning points have an assigned deadline.

2.3 Power plants and generating units

The power generation process is organized around geographical *locations*. Each location has to produce a minimum amount of electrical power to comply with government regulations. The power production in a certain location is the sum of the electrical power generated by several power plants. Each power plant has a specific location associated to it. Besides their location, power plants are also associated to divisions, for management purposes. Both associations, into locations and divisions, have an influence on the operation of power plants. This influence will be reflected on the problem restrictions.

A power plant hosts a certain number of *generating units*. Periodically, generating units may be halted for several reasons, including seasonal events and maintenance operations. The

execution of maintenance projects may be affected by restrictions on the simultaneous halting of generating units. For instance, once a generating unit goes down for maintenance, other units in the same power plant might have to be kept working, which might forbid the simultaneous execution of other maintenance projects on the remaining generators at that power plant. The impossibility of simultaneously executing certain maintenance projects establishes an exclusion dependency between maintenance projects. Only maintenance projects that cause the halting of generation units are affected by excluding dependencies.

3 Problem Input

The input data provided by AES-Tietê is composed by a set of projects, a set of warning points, a planning horizon, PH, the availability of resources along the PH, and other data about power plants and generating units.

The planning horizon is always taken to be five years, that is, its extent T is set to 60. However, the portfolio expected as a solution for the problem has an *execution horizon* (EH) of ten years ($2T$): the first five years are the planning horizon, while the final five years serve to accommodate projects that, although initiated before the final month on the PH, will finish after the end of the PH. With a duplication of the PH period we can register how much a project that terminates beyond the last month in the PH will contribute to reduce the portfolio's overall risk. Duplicating the planning horizon is enough, since no project execution is longer than five years.

3.1 Available resources

For administrative and fiscal reasons, and government regulations, resources must be classified into two categories: operational expenditures (OPEX) and capital expenditures (CAPEX). The costs of each project are always entirely of only one of these two categories.

Resource availabilities are presented in two series of five annual amounts. The first series designates annual resource availabilities for CAPEX expenditures, and the second one represents annual values for OPEX expenditures. Annual resource availabilities for years after the PH are replicated from the given resource availability for the first five years, in both categories.

3.2 Projects

A project is defined by the following set of parameters and constraints:

1. *Project identifier*: a unique, numeric identifier.
2. *Maintenance duration*: for maintenance projects, this parameter indicates whether the project's duration is *short-term* ('S') or *long-term* ('L'). For *non-maintenance* projects, the indication is 'N'.
3. An *area*: an identification of the administrative unit or power plant associated to a project. The administrative units include an "operation and management center" (COG), and an "information technology center" (IT). The power plants include: AGV, BAB, BAR, CAC, EUC, IBI, LMO, MOG, NAV, PRO, SJS, SJQ. See Table 1.

4. *Generating unit identifier*: for maintenance projects, the identifier of the generating unit in the corresponding power plant that will be affected by the project's execution. A maintenance project affects no more than one generating unit.
5. *Halting start month*: for a maintenance project, the halting start month indicates from which month, counting from its first execution month, the project will halt the corresponding generating unit.
6. *Halting duration*: for a maintenance project, the halting duration informs for how long, in months, the generating unit affected by the project will be halted.
7. A manually selected *initial month* at which the project is scheduled to start. The set of all projects, together with the corresponding initial months, forms a portfolio, dubbed the *initial portfolio*. This portfolio was manually constructed by expert engineers at AES-Tietê, and can be used as a yardstick for comparison against other solutions.
8. A project *lead time* is the number of months, counting from the beginning of the PH, after which the project can be scheduled to start.
9. A *project kind*, which can be *risk management* or *non-risk management*. The project kind is used to indicate whether a project is mandatory, and therefore must start at its assigned initial month. Non-risk management projects are mandatory.
10. A *resource classification* designating the kind of resource it consumes: OPEX or CAPEX. A project can consume resources from one category only.
11. A sequence of *values*, also referred to as the project's *costs*, describing the amount of resources the project will consume in each month during its execution, starting with the initial month. The *duration* of a project is, therefore, the number of values in its sequence of costs, from the first non-zero value up to the last non-zero value. Outside its duration, a project consumes no resources.

3.3 Warning points

Warning points are specified by the following parameters:

1. An *identifier*: a unique, numeric identifier.
2. A *risk measure*: a rational number indicating the risk associated to the warning point.
3. *Group of projects*: a list of project identifiers that defines the group of projects that controls this warning point. Note that some projects may not belong to any group and, therefore, they do not contribute to risk control, *e.g.*, a project for office supplies acquisition.
4. A *risk category*: a qualitative classification of the risk into two categories: *critical* or *non-critical*.
5. A *deadline*: the latest month at which the warning point must be completely controlled. The deadline is present for high-risk warning points only.

3.4 Power plants and generating units

The AES-Tietê power generation system comprises 12 power plants. Each plant hosts a certain number of generating units and must adhere to specific regulations for their operation, according to their geographical locations and administrative divisions.

Table 1 summarizes the administrative hierarchy of the AES-Tietê generation system, including the information about divisions, areas and power plants, geographical locations and the number of generating units in power plants.

The following parameters are provided for each power plant:

1. *Plant identifier*: a unique, three-letter code identifier;
2. *Generating units*: the number of generating units in the power plant;
3. *Division*: the division responsible for managing the plant;
4. *Geographical location*: the geographical location the plant resides in.

Table 1: Administrative hierarchy of the AES-Tietê system. The first column gives the administrative division the plant is subordinate to. The second column gives the area. The third column specifies the power plant location. The fourth column shows the number of generating units at the corresponding power plant.

Division	Area/Plant	Location	Generating Units
COG	COG	–	–
IT	IT	–	–
AGV	AGV	AV	6
PRO	NAV	NAPI	3
	PRO	NAPI	3
BAR	BAB	BBB	4
	BAR	BBB	3
	IBI	NAPI	3
LMO	CAC	RP	2
	EUC	RP	4
	LMO	RP	2
	MOG	RMG	2
	SJS	RJM	2
	SJQ	RJM	1

4 Constraints

In this section we list all constraints for this particular version of the PPS problem. Notice that this PPS version is not a linear problem, which entails that some of its constraints are expressed by logical statements; for instance, the halting restriction expressed in the Equation 13. This

does not pose a difficulty to our approach to handle the problem, since we adopt an heuristic technique, instead of using linear programming, to solve it.

Let I be the set of projects in an input instance for the PPS problem, let W be the set of warning points in that instance, and let T be the associated planning horizon, measured in months. We will always assume that T comprises an integral number of years, that is, it is a multiple of 12. For each project $p \in I$, we will indicate its duration, in months, by d_p . For ease of reference, we may write $[n]$ for the interval $[1, n]$, where n is a positive integer.

A portfolio P is a set of pairs (p, m) where p is a project in I and m is the month in which project p is scheduled to start its execution, that is, $P \subseteq I \times [T]$. We will indicate by $I_P \subseteq I$ the set of projects present in portfolio P .

A portfolio P is *valid* if it satisfies the following set of constraints.

Consistent scheduling. A project cannot be scheduled to start at two different months.

$$\text{If } (p, m_1) \in P \text{ and } (p, m_2) \in P \text{ then } m_1 = m_2, \quad \text{for all } p \in I_P. \quad (1)$$

This means that a valid portfolio P is in fact a function $P : I_P \mapsto [T]$. From now on, we will let $m_p = P(p)$ indicate the starting month and $e_p = m_p + d_p - 1$ will be the last execution month of project p , when $p \in I_P$.

Lead Time. A project p cannot be scheduled to start before its lead time l_p has passed. Then,

$$m_p > l_p, \quad \text{for all } p \in I_P. \quad (2)$$

Mandatory projects. A mandatory project p should always start at its prescribed initial month, p_m . Hence,

$$(p, m_p) \in P, \quad \text{for all mandatory projects } p \in I. \quad (3)$$

Intolerable warning points. A warning point whose risk category is ranked as high-risk is an intolerable warning point and have a deadline associated to it. Each intolerable warning point must be completely controlled by the end of its deadline. This means that all projects in the control group of any intolerable warning point must be terminated by that deadline. Let $G_w \subseteq I$ the group of projects controlling warning point w . Recall that W is the set of all warning points in I . Then we must have

$$G_w \subseteq I_P, \quad \text{for all intolerable warning points } w \in W. \quad (4)$$

This restriction being satisfied, define

$$f_w = \max_{p \in G_w} (m_p + d_p - 1). \quad (5)$$

We require, further, that

$$f_w \leq D_w, \quad \text{for all intolerable warning points } w \in W, \quad (6)$$

where D_w stands for the deadline associated to an intolerable warning point w .

Limited resources. Each project demands a certain amount of monthly resources from the month it starts until the last month of its execution. All resources consumed by a project are of the same kind as the resource classification associated to the project. The total yearly resource consumption of a certain kind, counting all projects that are under execution in that year, should not surpass the given amount of available resources of that kind and for that year.

The set of resource classifications is $Q = \{\text{CAPEX}, \text{OPEX}\}$. For any project $p \in I$, let $r_{p,m}$ be the resource consumption of project p during the m -th month of its execution ($1 \leq m \leq d_p$). For a portfolio P , we collect in $P_{t,q}$ the set of all projects of class $q \in Q$ that are active at month $t \in [2T]$, that is $P_{t,q} = \{p \in I_P \mid t \in [m_p, e_p]\}$. Then

$$c_{P,t,q} = \sum_{p \in P_{t,q}} r_{p,t-m_p+1}, \quad t \in [2T], q \in Q, P \text{ a portfolio}, \quad (7)$$

gives the resource consumption at month t for all active projects of class q that are already scheduled at portfolio P . We can now collect the consumption of resources of a class q in a year y for a portfolio P :

$$C_{P,y,q} = \sum_{t \in M_y} c_{P,t,q}, \quad y \in [1, T/12], q \in Q, M_y = [12y - 11, 12y], P \text{ a portfolio}. \quad (8)$$

So, if $S_{y,q}$ is the total amount of resources of class q available for year y , this restriction imposes the constraint

$$C_{P,y,q} \leq S_{y,q} \quad \text{for all } y \in [1, T/12], q \in Q, P \text{ a portfolio}. \quad (9)$$

We describe next the exclusion dependency constraints involving maintenance projects. We note that the only reason for halting a generating unit is the execution of a maintenance project that affects it. A maintenance project causes the halting of a generating unit for a limited period of time during its execution. That period is defined by the project's halting start month and halting duration.

Halting restrictions by geographical location. Restrictions involving the halting of generating units in some locations stem from engineering considerations. Let $g(u, m)$ denote the number of generating units halted at plant u and month m in a given portfolio, while $g_L(u, m)$ denotes the number of generating units halted at plant u and month m due to long-term maintenance projects. Referring to Table 1, the restrictions are as follows:

1. In the RP location, in any given month, if a plant has two or more generating units halted, the other plants cannot have any generating units halted. More formally, we have

$$(g(u_1, m) \geq 2) \rightarrow (g(u_2, m) = 0), \quad (10)$$

for $m \in [2T]$ and $u_1, u_2 \in \{\text{CAC}, \text{EUC}, \text{LMO}\}$, with $u_1 \neq u_2$.

2. In the BBB location it must be observed that:

- In any given month, if a plant has two or more generating units halted, the other plants cannot have any generating units halted. Thus,

$$(g(u_1, m) \geq 2) \rightarrow (g(u_2, m) = 0), \quad (11)$$

for $m \in [2T]$ and $u_1, u_2 \in \{\text{BAB}, \text{BAR}\}$, with $u_1 \neq u_2$.

- In any given month, at most two generating units from the BAR plant can be down. Thus,

$$g(BAR, m) \leq 2, \quad \text{for } m \in [2T]. \quad (12)$$

- In any given month, two generating units from the BAB plant can be down simultaneously, but only if none of the BAR plant generating units are halted. Thus,

$$(g(BAB, m) = 2) \rightarrow (g(BAR, m) = 0), \quad \text{for } m \in [2T]. \quad (13)$$

3. In the NAPI location, at most two generating units can be down at any given month. Thus,

$$g(NAV, m) + g(PRO, m) + g(ABI, m) \leq 2, \quad \text{for } m \in [2T]. \quad (14)$$

4. In any given month, in the AV location, we can have at most two halted generating units due to long-term maintenance projects. Thus,

$$g_L(AGV, m) \leq 2, \quad \text{for } m \in [2T]. \quad (15)$$

Halting restrictions at the AGV plant. In any given month, if two or more generating units from the AGV plant are down, then no generating unit from the BAB, NAV, or PRO plants can be halted, that is

$$(g(AGV, m) \geq 2) \rightarrow (g(BAB, m) = g(NAV, m) = g(PRO, m) = 0), \quad \text{for } m \in [2T]. \quad (16)$$

Halting restrictions by division. In any given month, no more than three generating units in the same division can be halted. Let D be the set of divisions in the AES-Tietê electric power generation system, as given by Table 1. For each division $d \in D$, let U_d be the set of power plants in d . We must have

$$\sum_{u \in U_d} g(u, m) \leq 3, \quad \text{for } m \in [2T], d \in D. \quad (17)$$

For instance, if no other constraint takes effect, then we are allowed to halt one generating unit from each of the BAB, BAR, and ABI power plants, in the BAR division.

In a practical scenario, real problems of this complexity might involve uncertainties. Although some recent work [14, 12] expands on this issue, the lack of trustworthy data from AES-Tietê prevented a more detailed treatment at this phase of the present modeling.

5 Objective function

A set of optimization goals can be proposed for the PPS problem depending on which aspect — costs, human resources, risks — one wants to focus upon. While planning and constructing a portfolio, AES-Tietê decision makers focus their efforts on reducing the risk resulting from all uncontrolled warning points. For that, they try to judiciously select and schedule projects of the risk management kind. The objective function of the PPS problem studied here should also reflect the best choice of scheduling projects in order to guard against uncontrolled warning

points. So, in this version of the PPS problem, we focused on the optimization goal of *reducing the risk of uncontrolled warning points* in a portfolio.

As we advance through the EH time frame, projects terminate and warning points reach a controlled state. When that happens, the total stock of unguarded risks decreases until it reaches a minimum, ideally 0, at month $2T$ or earlier. The area under the risk curve summarizes in a single number two of the most relevant aspects of a portfolio: the *amount* of controlled risk and *when* the risk is averted. That is, the earlier a warning point is effectively controlled, and the larger is the risk value associated to the warning point, the better is the corresponding portfolio. Therefore, we set the objective function to minimize the area under the risk curve associated to the portfolio.

Recall that W is the set of all warning points in an input instance I to the PPS problem. Let R_w denote the risk associated to warning point $w \in W$. For a portfolio P we let $W_P \subseteq W$ be the set of warning points that are effectively controlled in P . Then, the area under the risk curve for P is

$$R(P) = 2T \sum_{w \in W} R_w - \sum_{w \in W_P} R_w(2T - f_w), \quad (18)$$

where f_w , given by Eq. (5), is the first month when warning point w is finally controlled.

For a given input instance I to the PPS problem, let V_I be the set of all viable portfolios that can be obtained from I . The goal, then, is to construct a portfolio P^* such that $R(P^*)$ is minimum among $\{R(P) \mid P \in V_I\}$. In the next section we describe a heuristic strategy that searches for a portfolio P that, although not guaranteed to attain $R(P) = R(P^*)$, improves considerably its risk measure when compared to other portfolios manually constructed by expert engineers.

6 A GRASP Variant for the PPS Problem

We describe a variant of the GRASP meta-heuristic for finding solutions for the PPS problem treated here that are better than the manually produced solutions provided by specialists. GRASP is a multi-start meta-heuristic that generates good quality solutions for many combinatorial optimization problems [21, 8], besides being of fast implementation, being easily comprehensible, and not requiring restrictive assumptions of mathematical optimization models; e.g. linear constraints. Thus, the GRASP meta-heuristic is suitable as a first approach to try and find better solutions than the manual ones.

The algorithm repeats two phases: (i) A construction phase in which a feasible solution is obtained; and (ii) a local search phase in which the algorithm finds a local optimal solution by iteratively checking whether there is a better solution in a neighborhood of the current solution. Several initial feasible solutions are constructed and improved upon until the best overall solution is then chosen. We implemented a variant of this basic strategy. First, a pool of viable portfolios is constructed. Then, we apply the local search procedure to the β best solutions in the pool, where β is a fraction of the pool size. This cycle is repeated a number of times, and the best solution is finally returned.

The pseudo-code for the GRASP routine is presented as Algorithm 1. In the next two subsections the construction and local search phases are discussed more fully. The choices of parameters appearing in the algorithms are discussed in more detail in Section 7.

Algorithm 1 GRASP-based routine for the PPS problem

```

1. Initialize an empty portfolio as the current BEST_SOLUTION.
2. for  $R$  iterations do
3.   // Construction phase
4.   Initialize an empty pool of solutions  $\Pi$ .
5.   while  $|\Pi| < C$  do
6.     Construction procedure returns a solution  $P$  for the PPS problem.
7.     if  $P$  satisfies the intolerable warning point constraint then
8.       Include  $P$  into the pool  $\Pi$  of solutions.
9.     end if
10.  end while
11.  // Local search phase
12.  for Each solution  $P$  in the first  $\beta$  best solutions in  $\Pi$  do
13.    Let  $P = \text{LOCAL\_SEARCH}(P)$ 
14.    if  $P$  is better than BEST_SOLUTION then
15.      BEST_SOLUTION =  $P$ 
16.    end if
17.  end for
18. end for
19. return BEST_SOLUTION.

```

6.1 The Construction Phase

The first phase in our GRASP-based algorithm consists in the repeated execution of a construction procedure. As shown in Algorithm 1, the construction procedure produces a solution and checks whether it satisfies the intolerable warning point constraint as expressed by Eqs. (4) and (6). These constraints require that all projects that control warning points must be scheduled in such a way that the corresponding warning points are totally under control before their deadlines are reached. If the portfolio is valid, then it is included in the pool of constructed solutions; otherwise, it is discarded. Feasible solutions are inserted into the pool until its capacity C is reached. Notice that the same solution might be included into the pool more than once. This does not seem to be a problem, however, and, moreover, since the solutions are relatively large structures, checking for equality is costly and does not seem to be worth the effort. The best β of the constructed solutions in the pool are chosen as input solutions for the local search phase.

Two kinds of construction procedures were considered: an adaptive one and a non-adaptive one. We start with the non-adaptive procedure.

6.1.1 The non-adaptive procedure

The non-adaptive heuristic was devised as a simple procedure using two candidate lists of pairs (p, m) of projects $p \in I_P$ and potential starting months $m \in [l_p + 1, \min(MD_w - d_p + 1, T)]$, where MD_w is the minimum among all deadlines associated to all intolerable warning points p helps to control. If a project $p \in I_P$ does not control any warning points, we consider all pairs (p, m) with $m \in [T]$. The lists are then sorted according to a suitable key parameter. The key parameter associated to a pair (p, m) should measure the gains obtained by scheduling project p at month m , based not only on p 's ability to control risk, but also on its total cost. Notice that a project with a large value of controlled risk may be less cost-effective in terms of controlled risk per unit of cost than another project [18]. Notice also that we use two candidate lists instead of a single list, as is standard in the construction phase of a GRASP routine. This allowed us to classify the projects into two priority categories.

Let I be the set of projects in an input instance for the PPS problem, with planning horizon

T , and warning point set W . In order to assess the cost-effectiveness of a pair $(p, m) \in I \times [T]$, we compute $b_{p,m}$, the *benefit* of the pair (p, m) . Let R_p be the sum of warning point risk shares project p contributes to control, computed over all warning points in W , and recall that $r_{p,t}$ gives the resource consumption of project p during its t -th month of execution. We let

$$b_{p,m} = R_p \left((2T - m - d_p + 1) / \sum_{t=1}^{d_p} r_{p,t} \right), \quad (19)$$

so that $b_{p,m}$ expresses the amount of risk project p helps to control, weighted by how long that contribution is effective per unit of cost.

The portfolio construction procedure initially pre-processes the input data to schedule all mandatory projects in I , which are then removed from I . Next, all pairs $(p, m) \in I \times [T]$, have their benefit calculated, and the two candidate pair lists are built and sorted in non-increasing order of benefit. The *high-risk sorted pairs* (HRSP) list involves pairs (p, m) where project p contributes to control high-risk warning points. The *sorted pairs* (SP) list involves all the remaining pairs in $I \times [T]$. We split the standard GRASP candidate list in two parts in order to give pairs in the HRSP list a higher priority over the SP list, since the corresponding projects help control high-risk warning points.

The algorithm proceeds to construct a portfolio by iteratively choosing a pair (p, m) from one of the two candidate lists. The next pair is chosen from the HRSP or the SP lists with probabilities η and $1 - \eta$, respectively. Once the list to choose the next pair from is determined, the next pair (p, m) is randomly chosen from among the first k pairs in that list. Then pair (p, m) is included in the portfolio being constructed if, by including it, the new portfolio does not violate any of the following *local constraints*:

- “consistent scheduling”, Eq. (1);
- “lead time”, Eq. (2);
- “limited resources per year”, Eq. (9); and
- “halting restrictions”, Eqs. (10–17).

Observe that, at each iteration, as a new pair (p, m) is included, the available resources for each year decrease, reflecting the resource usage of project p , and the cycle repeats until no pair (p, m) can be included in the portfolio.

At the end of the construction procedure the portfolio generated is returned.

We summarize the non-adaptive procedure used in the construction phase as Algorithm 2.

6.1.2 The adaptive procedure

An adaptive version of the construction procedure involves the recalculation of the benefits of each remaining pairs in the candidate lists, after a new pair is included in the portfolio being constructed. After recalculation, the lists are sorted again.

The benefit function given by Eq. (19) is a first attempt to express the cost-effectiveness of scheduling a project at a certain month. However Eq. (19) does not take into account the impact on available resources caused by scheduling a new project. Given a portfolio P , we want to measure the impact of including a new pair (p, m) in P by estimating which fraction

Algorithm 2 Non adaptive construction procedure for the PPS problem

-
1. Initialize P as an empty portfolio.
 2. **for all** mandatory projects **do**
 3. Schedule each mandatory project at its initial scheduled month.
 4. Subtract mandatory project's resources from the available resources.
 5. **end for**
 6. Calculate the risk R_p of each remaining project p .
 7. Construct the HighRiskSortedPairs (HRSP) list of high risk project \times month pairs, sorted by benefit value.
 8. Construct the SortedPairs (SP) list of remaining project \times month pairs, sorted by benefit value.
 9. **while** HRSP is not empty or SP is not empty **do**
 10. If one of the lists is empty, choose the other one, otherwise, randomly choose between HRSP and SP with probabilities η and $1 - \eta$, respectively.
 11. Get a pair (p, m) randomly chosen from the first k pairs at the top of the chosen list.
 12. Remove pair (p, m) from the chosen list.
 13. **if** (p is not already scheduled) **and** (adding (p, m) to the solution does not violate local constraints) **then**
 14. Let $P = P \cup \{(p, m)\}$.
 15. **else**
 16. Discard pair (p, m) .
 17. **end if**
 18. **end while**
 19. // Any remaining projects in SP and HRSP are not scheduled.
 20. Return P .
-

of the presently available resources a new pair (p, m) will consume, at each month project p is in execution. Furthermore, since costs are grouped annually, we need to sum p 's resource consumption over all months it is active in each year.

We denote by $M(p, m, y)$ the set of months project p will be active during year y , if it starts at month m . Note that, in this case, p would end its execution at month $e_p = m + d_p - 1$. Note also that $12(y - 1) + 1$ is the first month of year y , and $12y$ is its last month. We then have, for $y \in [1, 2T/12]$:

$$M(p, m, y) = \begin{cases} \emptyset, & \text{if } 12y < m \text{ or } e_p < 12(y - 1) \\ [m, \min\{e_p, 12y\}], & \text{if } 12(y - 1) < m \leq 12y \\ [12(y - 1) + 1, \min\{e_p, 12y\}], & \text{if } m \leq 12(y - 1) < \min\{e_p, 12y\}. \end{cases} \quad (20)$$

Then, if $h(p, m, y)$ gives the resource consumption of project p during year y when started at month m , we have

$$h(p, m, y) = \begin{cases} 0, & \text{if } M(p, m, y) = \emptyset \\ \sum_{j \in M(p, m, y)} r_{p, j-m+1}, & \text{otherwise.} \end{cases} \quad (21)$$

Recall that, by Eq.(8), $C_{P, y, q}$ gives the total resource consumption for projects of category q in a portfolio P during year y , and $S_{y, q}$ is the total amount of resources of category q in a year y , as given by the input instance. Then,

$$\kappa(p, m, y) = \frac{h(p, m, y)}{S_{y, q} - C_{P, y, q}}, \quad (22)$$

gives the fraction of the currently available resources, at year y , that will be consumed by the inclusion of a pair (p, m) in the current portfolio P , where q is p 's category. To avoid division by zero, when $S_{y, q} = C_{P, y, q}$, we set $\kappa(p, m, y) = 0$.

The maximum fraction of yearly resource usage that a project p will incur when the pair (p, m) is considered for inclusion in a portfolio P is, then,

$$\rho(p, m) = \max_{1 \leq y \leq T/12} \kappa(p, m, y). \quad (23)$$

This same idea can be used with other functions instead of the ‘maximum’ in Eq. (23), *e.g.*, ‘minimum’, ‘average’, or ‘sum’.

In order to recalculate the cost-effectiveness of a project p being scheduled at month m in a portfolio P , and now taking into consideration the changes in available resources, we adopted as a measure of its effectiveness its adaptive benefit $b_a(P, p, m)$, defined as

$$b_a(P, p, m) = R_p \left((2T - m - d_p + 1) / \rho(p, m) \right). \quad (24)$$

We note that every project consumes a non-null amount of resources in its corresponding category. The adaptive construction procedure is shown in Algorithm 3.

Algorithm 3 Adaptive construction procedure for the PPS problem

1. Initialize P as an empty portfolio.
 2. **for all** mandatory projects **do**
 3. Schedule each mandatory project at its initial scheduled month.
 4. Subtract mandatory project’s resources from the available resources.
 5. **end for**
 6. Calculate the risk R_p of each remaining project p .
 7. Construct the HighRiskSortedPairs (HRSP) list of high risk project \times month pairs, sorted by adaptive benefit.
 8. Construct the SortedPairs (SP) list of remaining project \times month pairs, sorted by adaptive benefit.
 9. **while** HRSP is not empty or SP is not empty **do**
 10. If one of the lists is empty, choose the other one, otherwise, randomly choose between HRSP and SP with probabilities η and $1 - \eta$, respectively.
 11. Get a pair (p, m) randomly chosen from the first k pairs at the top of the chosen list.
 12. Remove pair (p, m) from the chosen list.
 13. **if** (p is not already scheduled) **and** (adding (p, m) to the solution does not violate local constraints) **then**
 14. Let $P = P \cup \{(p, m)\}$.
 15. **else**
 16. Discard pair (p, m) .
 17. **end if**
 18. Recalculate the adaptive benefit of each remaining pair in SP and HRSP.
 19. Reorder SP and HRSP lists according to the new adaptive benefit values.
 20. **end while**
 21. // Any remaining projects in SP and HRSP are not scheduled.
 22. Return P .
-

6.1.3 Comparing the construction strategies

We performed some experiments in order to compare the adaptive and the non-adaptive construction procedures. We used as input instance the real-world based data described in Section 7.

In order to check how much the resource usage has an impact on the quality of solutions, we performed experiments using variations on the adaptive benefit $\rho(p, m)$ described by Eq. (23), namely, we used the minimum, the average and the sum of the fractions of resource usages. We also used a pool with a capacity C of 100 solutions and also recorded the number of unfeasible solutions obtained until we reached full capacity. Table 2 shows the result of these

Table 2: Construction strategy, best solution, average solution value (both in 10^6 risk-month units), number of solutions, running time values (in seconds) for a real-world input instance.

Strategy	resource usage	best OOF	avg OOF	unfeasible solutions	Time (s)
Non adaptive		0.570	0.638	0	74
Adaptive	avg	0.645	0.698	118	2343
	max	0.741	0.846	43	1417
	min	0.739	0.820	24	932
	sum	0.649	0.700	128	1691

experiments, where the real instance was used as input to the algorithm. The first column defines the construction strategy, the second column specifies how the fraction of yearly resource consumptions are combined, the next two columns give the best and the average values of the optimized objective function (OOF) values, respectively, the last two columns report the number of unfeasible solutions that were constructed and the running times, respectively.

From Table 2, we find that the best solution found by the adaptive procedure – using the maximum resource usage in the benefit function – is around 23.4% worse than the one found by the non-adaptive procedure. In addition, the adaptive procedure finds more invalid solutions and takes considerably more time to be executed than the non-adaptive strategy. Given these limitations on the side of the adaptive procedure, we decided to adopt the non-adaptive procedure in our GRASP-based heuristic.

6.2 Local search

The second phase of our GRASP heuristic consists of executing a local search procedure. For a fraction of the solutions obtained in the construction phase, a distinct local search procedure is executed. The local search starts at the input solution and examines its neighborhood, checking whether there is a better solution in terms of the objective function, using a best improvement strategy. It iteratively repeats the search in the neighborhood of each new solution until no better solution is found. The best solution thus found is then returned. The local search procedure is shown as Algorithm 4.

Algorithm 4 Local search procedure

1. Let P be the initial solution.
 2. **repeat**
 3. Get the list of neighbors N of the current solution P .
 4. Search through N , using a best improvement strategy.
 5. **if** an improved solution Q was found **then**
 6. Let $P = Q$
 7. **end if**
 8. **until** There is no improved solution for the current solution P .
 9. **return** Portfolio P .
-

A fundamental aspect of a local search procedure concerns how the neighborhood of a solution is defined. We defined the neighborhood of a portfolio as the set of portfolios where each neighbor is obtained by re-scheduling one of its projects. If a project was originally scheduled at month m , a neighbor is obtained by re-scheduling it at months from $m - \Delta$ to $m + \Delta$, avoiding the current m , and where Δ is a fixed parameter. If re-scheduling leads to infeasibility, or to a project starting at a month beyond T , that neighbor is discarded.

We note also that only projects whose start times are in the interval $[1, T]$ are considered for re-scheduling. Another relevant aspect of the local search is how the neighborhood is searched. We implemented a best improvement strategy, which searches through all projects in the neighborhood and selects the best one. Algorithm 5 shows the procedure to construct the neighborhood of a portfolio.

Algorithm 5 Returns the neighborhood of a solution

1. Let P be the initial solution.
 2. Initialize an empty list of solutions N .
 3. **for all** pairs (p, m) in P **do**
 4. **for all** months t in the interval $[m - \Delta, m + \Delta] - \{m\}$ **do**
 5. $Q = (P - \{(p, m)\}) \cup \{(p, t)\}$
 6. **if** Q is feasible **then**
 7. Let $N = N \cup \{Q\}$
 8. **end if**
 9. **end for**
 10. **end for**
 11. **return** List of neighbors N
-

7 Experiments

We performed experiments to evaluate the performance and robustness of the complete algorithm. As solution quality indicators, we chose the heuristic running-time and the objective function value.

In the AES-Tietê input instance, we had 1411 projects for a planning horizon comprising the years 2013–2017. The total number of warning points were 434 and the sum of all attributed risk values was around 25×10^3 risk-units. The annual costs were 1.5, 1.0, 1.7, 1.3, and 0.7, expressed in 10^9 monetary units. These figures were multiplied by an undisclosed constant, in order to comply with AES-Tietê confidentiality policies.

The amount of available yearly CAPEX resources was obtained by summing up the CAPEX costs of all projects in the initial manual solution that were active at each year. Yearly OPEX available resources were obtained in a similar fashion.

7.1 Input instances

In order to assess the indicator values and test the robustness of the heuristic, we compared the manual portfolio solution for a real-world instance against the solution generated by the algorithm for that same instance. The manual portfolio solution was built by expert managers and engineers at AES-Tietê.

In order to assess the robustness of the procedure, we used the real-world instance provided by AES-Tietê as a seed for generating similar instances, and randomly disturbed the seed instance data, in a controlled way and by small margins. The disturbing procedure applied the following modifications to the real instance data, where d was a fixed disturbance factor:

Project costs: modified by $x\%$, with x randomly chosen between $-d$ and $+d$.

Warning point risks: modified by $x\%$, with x randomly chosen between $-d$ and $+d$.

Other parameters: not modified in the disturbed instances.

Table 3: Probability of choosing HRSP list (η), best and average solution values (OF, for objective function; in 10^6 risk-month units), number of unfeasible solutions (Unfeasible), and running time values (Time) in seconds.

η	Best OF	avg OF	Unfeasible	Time
20	0.57182	0.63542	397	116
25	0.57155	0.64131	72	89
30	0.56832	0.64124	41	87
35	0.57327	0.64124	9	84
40	0.57392	0.64086	4	84
45	0.57244	0.64014	2	85
50	0.57702	0.64172	4	86
55	0.56897	0.64071	5	86
60	0.56927	0.64042	6	84
65	0.56780	0.64067	4	85
70	0.57194	0.64060	3	86
75	0.56850	0.63603	2	86
80	0.57321	0.63434	0	87
85	0.57562	0.63427	0	86
90	0.56814	0.63477	0	87
95	0.56832	0.63560	1	98
100	0.57046	0.63316	0	115

Using these modifications, we generated 10 instances for a 5% disturbance factor. We decided to adopt the disturbance factor of 5% because, according to engineers, modifications in the overall cost beyond 5% in a year are unlikely in a traditional industry such as in the electrical power generation industry, whose assets do not change by a large amount yearly.

7.2 Heuristic parameters

Before running the robustness experiment, we performed some calibration experiments in order to identify adequate values for some of the GRASP parameters.

During the construction phase, the procedure chooses between the HRSP and the SP lists with probabilities η and $1 - \eta$, respectively. In order to choose a proper value for η , we executed the construction procedure until it obtained 1000 feasible solutions for each η value. Table 3 shows the best and average objective function values, the number of unfeasible solutions constructed until 1000 feasible solutions are reached, and the running time, for selected η values. Using the experiment's results, we set $\eta = 60\%$ since this yields very good values for both best and average solution values, as well as gives rise to few unfeasible solutions and shows good running time figures. This choice increases the odds that a high-risk controlling project will be scheduled earlier in the portfolio.

While the construction procedure was producing a new portfolio, the next project and month pair, (p, m) , to be inserted in the current partial solution was randomly chosen among the first k best candidates in the sorted candidate list. We set k equal to 5, because this value was a good compromise between enforcing the randomized aspect of the heuristic while not losing the greedy advantage of having the candidate list sorted.

Table 4: Best, average, and input instance solution values (in 10^6 risk-month units), and running time values (in seconds) for distinct input instances. The last column contains total remaining resource values (in 10^9 monetary units) for the best solution.

DF - instance	min OOF	avg OOF	IOF	Time	remaining resource
0%	0.54682	0.58865	0.99631	285.506	0.328
5% - a	0.51916	0.56113	0.99907	129.133	0.252
5% - b	0.52152	0.55443	0.97661	131.456	0.259
5% - c	0.51454	0.55913	0.98841	136.996	0.285
5% - d	0.54014	0.57090	0.98184	139.128	0.271
5% - e	0.52945	0.56467	0.97402	131.480	0.279
5% - f	0.51884	0.55234	0.98141	149.432	0.254
5% - g	0.53923	0.57053	0.99235	134.487	0.251
5% - h	0.54274	0.58001	0.99567	143.270	0.270
5% - i	0.51250	0.56150	0.97650	132.105	0.334
5% - j	0.52402	0.56475	0.98433	135.379	0.255

We performed several experiments (results not shown) to decide on the best values for parameters C , the solution pool size, and β , the number of solutions to carry on to the search phase. Since the local search phase is the most time consuming computation performed by the heuristic, it proved to be an advantage to use $\beta < C$. The experiments took as input the real instance, as well as the disturbed instances described in Section 7.1. We finally decided on 200 for the pool capacity C and set β to be 10% of the capacity.

For the local search phase, we experimented with distinct values for the parameter Δ , used in finding neighbor solutions. We found empirically that setting Δ to 8 covers a good amount of neighbor solutions without compromising the algorithm's performance.

7.3 Results and analysis

The GRASP based heuristic was executed once for each input instance, and the best solution was retained. The experiments were performed in an Intel[®] i3 Dual Core[®] HT 1333GHz CPU, with 4Gb of RAM, running the Ubuntu GNU/Linux distribution, version 12.04 LTS, with kernel 3.5.0. The algorithm was implemented using the GO 1.1.2 programming language for its enhanced performance and ease of coding. In order to improve the algorithm's performance, both the construction and local search procedures were implemented using GO parallel programming features.

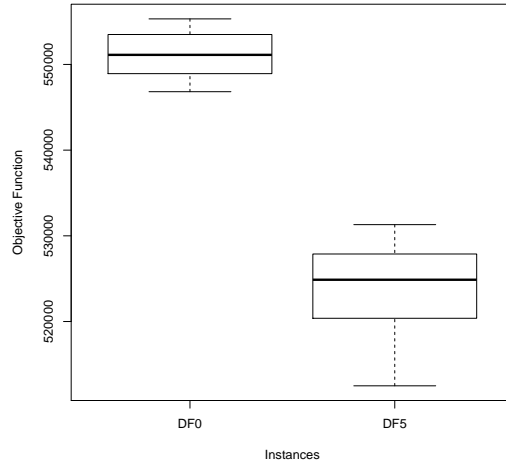
The results are summarized in Table 4. The first column shows the disturbance factor (DF) and an instance identification. A DF of 0% refers to the real instance. The second and third columns are, respectively, the best and the average optimized objective function (OOF) values in risk-month units. The next column gives the objective function value for the corresponding manual portfolios (IOF) in risk-month units. The last column informs the total running time (Time) for executing one run for each instance, in seconds.

The results show that the heuristic consistently found solutions whose objective function values were around 47% better when compared to the corresponding values for the manual, undisturbed solutions. This behavior was repeated when the algorithm took as input any of the disturbed instances, attesting to its robustness when relatively small changes were applied

to the real input instance.

Figure 1 shows some optimized (OOF) values statistics for two typical instances. The data is summarized in two boxplots: the first one presents statistics for the OOF values (in 10^6 risk-month units) obtained from the real instance (DF0). The second boxplot presents the OOF statistics for a 5% perturbed instance (DF5).

Figure 1: Best solution value statistics for the real instance and a perturbed instance.



Instance	Min	1st Q	Median	Mean	3rd Q	Max
DF0	0.54682	0.56629	0.57926	0.58865	0.59529	0.70257
DF5	0.51250	0.53731	0.54994	0.56150	0.56828	0.70926

Besides the objective function value, we also obtained the total amount of remaining resources after each portfolio was constructed, in order to check the impact of the heuristic on resource consumption. Table 4 contains a summary of the total remaining resources (in 10^9 monetary units) along all the PH, and per input instance. The first line corresponds to the undisturbed instance, and each of the subsequent lines gives the result for the corresponding disturbed instance, as described in Section 7.1.

8 Conclusions

We modeled a project portfolio selection (PPS) problem derived from a large electrical power generation and distribution company, namely, AES-Tietê, operating in Brazil. Besides traditional obstacles, such as resource constraints, the problem also involves complex features, such as elaborate exclusion dependencies among projects and an intricate relationship among projects that control warning points.

We proposed a variant of the GRASP meta-heuristic for solving this specific PPS problem. We modified the construction phase by using a greedy randomized procedure to construct feasible solutions that are stored in a pool of solutions. First, the greedy procedure used two candidate lists. Secondly, only a fraction of the best solutions in the pool are fed into the local search phase, which uses a best improvement strategy. Neighbor solutions used in the local search procedure are defined by changing a project's scheduled time by a small amount.

We performed experiments using real-world input instances in order to find appropriate parameters for the heuristics and to assess the algorithm robustness. The heuristics showed a significant improvement over the manual solutions when treating instances of real data, as well as when taking disturbed data as input.

Future work will involve the inclusion of new constraints concerning different categories of available resources, such as human resources, the concurrent use of multiple objective functions, and versions of the problem where projects can be made available, or be dismissed, in a dynamic fashion. The heuristic will include more sophisticated adaptive procedures in the construction phase, as well as, improvements in the performance of the local search by adopting different neighborhood search strategies.

Acknowledgment

The authors would like to thank AES-Tietê for their technical support and real data sets, and thank the National Agency for Electrical Energy (ANEEL) for their financial support (grant PD 0064-1021/2010).

References

- [1] 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.
- [2] N.P. Archer and F. Ghasemzadeh. A decision support system for project portfolio selection. *International Journal of Technology Management*, 16(1):105–114, 1998.
- [3] N.P. Archer and F. Ghasemzadeh. An integrated framework for project portfolio selection. *International Journal of Project Management*, 17(4):207–216, 1999.
- [4] N. Baker and J. Freeland. Recent advances in R&D benefit measurement and project selection methods. *Management Science*, pages 1164–1175, 1975.
- [5] J.M. Campbell and R.A. Campbell. *Analyzing and managing risky investments*. CPI, 2001.
- [6] 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.
- [7] SC Chiam, KC Tan, and A. Al Mamum. Evolutionary multi-objective portfolio optimization in practical context. *International Journal of Automation and Computing*, 5(1):67–80, 2008.
- [8] P. Festa and M.G.C. Resende. GRASP: basic components and enhancements. *Telecommunication Systems*, 46(3):253–271, 2011.
- [9] 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.

- [10] 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.
- [11] 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.
- [12] Willy Herroelen and Roel Leus. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165(2):289–306, 2005.
- [13] J.W. Lee and S.H. Kim. Using analytic network process and goal programming for interdependent information system project selection. *Computers & Operations Research*, 27(4):367–382, 2000.
- [14] Juuso Liesiö, Pekka Mild, and Ahti Salo. Robust portfolio modeling with incomplete cost information and project interdependencies. *European Journal of Operational Research*, 190(3):679–695, 2008.
- [15] M. Martinsuo and P. Lehtonen. Role of single-project management in achieving portfolio management efficiency. *International Journal of Project Management*, 25(1):56–65, 2007.
- [16] 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.
- [17] C Mira, P Feijão, M A Souza, A Moura, J Meidanis, G Lima, R Bossolan, and I T Freitas. A project portfolio selection decision support system. In *Service Systems and Service Management (ICSSSM), 2013 10th International Conference on*, pages 725–730. IEEE, 2013.
- [18] C Mira, P Feijão, MA Souza, A Moura, J Meidanis, G Lima, R Bossolan, and I Freitas. A GRASP-based heuristic for the Project Portfolio Selection Problem. In *The 15th IEEE International Conference on Computational Science and Engineering (IEEE CSE2012)*, volume 1, pages 36–41. IEEE, 2012.
- [19] 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.
- [20] M. Rabbani, M. Aramoon Bajestani, and G. Baharian Khoshkhou. A multi-objective particle swarm optimization for project selection problem. *Expert Systems with Applications*, 37(1):315–321, 2010.
- [21] Mauricio G. C. Resende. Greedy Randomized Adaptive Search Procedures. In *Encyclopedia of Optimization*, pages 1460–1469. Springer, 2009.