Universidad de la República
Facultad de Ingeniería
Instituto de Matemática y Estadística
Prof. Ing. Rafael Laguardia (IMERL)
Montevideo – Uruguay

Tesis de Maestría
en
Ingeniería Matemática

POWER GENERATION INVESTMENT: AN ALGORITHM TO GET A GOOD FEASIBLE SOLUTION FROM OPERATION SIMULATIONS

Director de tesis: Dr. Alejandro Jofré
ajofre@dim.uchile.cl

Autor: Ing. Daniel Tasende
dtasende@ute.com.uy

Diciembre 2002

POWER GENERATION INVESTMENT: AN ALGORITHM TO GET A GOOD FEASIBLE SOLUTION FROM OPERATION SIMULATIONS

D. Tasende - dtasende@ute.com.uy

Abstract

The algorithm we present here is able to give an optimal solution for the power generation expansion problem if there is no dynamical links between time steps (normally years). In other case, it gives a good approximation to the optimal solution. It uses as input an estimation of the expected operation cost with fixed generation equipment. This kind of data is normally available in Power Utilities, frequently from operation models. For the general problem (solved with more involved methods), this algorithm can give a good feasible solution that can be useful in the search for the optimum.

1. Introduction

Electricity wholesale markets are changing their regulatory frameworks in a varied way. Depending on the case, different agents can be responsible for a safe demand supply, including the final consumer. In any case, there is an agent that will have a high cost if contracted generators would not be able to supply part of his demand.

Let us define a purchaser as some agent (private or public, final consumer or retailer) that somehow has the responsibility of a safe supply. The following examples are possible:

- The regulatory authority could be a purchaser for all the system's demand, with an implicit contract with all the power generators in the system. In this context the cost for not serving energy is the not served energy social unit cost, that is frequently represented as a stepwise linear function.
- A concessionaire in a specific geographical area could be a purchaser for his area. Now the cost for not serving energy is the penalty that the concessionaire has to pay in case of shortage. In this case the concessionaire will transfer the penalty to the contracted generators.
- Suppose a retailer with a set of supply contracts with consumers. Contracts could have different penalties, producing a convex stepwise linear cost function, as in the case of the regulatory authority.

The above mentioned purchasers correspond to different regulatory situations, but the structure of the problem is the same for all of them.

We are interested in the case in which the purchaser is able to install power generation. This is not permitted in some regulatory frameworks, nevertheless long term contracts are similar to buying a power plant (we are really interested in irreversibility).

The statement of the purchaser problem is to minimize a cost function including penalties in critical events, and we have a finite number of real investment alternatives. In real life, alternatives can include power contracts too. Long term power contracts could be included as alternatives in the present context. It is necessary that no long term contract could expire before the end of the study period.

When modeling oligopoly markets, game theory (see [2]) consideration may be necessary. The analysis of the whole market equilibrium can become an involved problem. The simplified statement of the investment problem that we present here could help to manage such kind of problem.

If we are studying real investments (or long term contracts), once the purchaser have invested, all the other players in the market can know his decision and are able to change their own investment decisions or offers in the Spot market and short term contracts market.

The main idea is to make simplifying assumptions in the statement of the problem in such a way that stochastic dynamic programming could be used. In the deterministic case, the number of transitions is further reduced using additional conditions on the problem.

Our problem is to choose in a set of **ip** power projects. A feasible solution will be a combination of the available power projects for each year in a period, such that each year has not more installed projects than the following year. Alternatives are combination of the **ip** possible power projects. The maximum number of alternatives is equal to the number of sub-sets in the set of possible projects ($\mathbf{2^{ip}}$). In real life problems there are many forbidden combination of projects. For example, if **k** projects are identical, they produce only **k** alternatives. There are complete system problems having a low number of possible projects (5-10 in the system of Uruguay). The first condition to be able to apply this method is to have a low number of alternatives (up to 1000 for a PC).

Even if we can have a great number of scenarios, they must be constructed as sequences of a few situations. In some problems we can construct scenarios from only two or three system annual situations (say low, average and high). Moreover, given a situation and an alternative for the corresponding year, it must be possible to estimate accurately the conditional expected annual operation cost.

When the simplifying assumptions above are valid, we are able to solve the problem using the algorithm we develop in this paper. The optimal solution is a strategy, defining the decision to be made in the first time step, and decisions for each bundle of scenarios at each time step.

Section 2 of this study is devoted to the deterministic version of the algorithm.

The assumption developed in 2.2 is needed to be able to use the algorithm, at least as an approximation to the solution of the problem. It has been tested for the system of Uruguay.

Result in 2.3 means that if in a time step, an alternative is more expensive in total costs than another and has less investment costs, then its investment is not enough for the system, and this situation will last with system growing (normally in the future). This result is only used to reduce the number of transitions to be considered. It depends on a monotony condition (MC) that is proved for a simplified problem.

In 2.4 we present a first formulation of the problem as a 0-1 non linear program. This is the basic formulation, close to the real life problem, but it is not well suited for resolution.

In 2.5 we present an equivalent linear formulation that enables the use of available methods and packages. See the chapter of integer programming in [3] for a survey on equivalent systems. In this context “equivalent linear 0-1 program” means a linear program with the same solution set as the basic formulation.

Linear problems of that kind are widely studied in the literature. Methods have been developed to reduce calculation time. Good routines have been developed to implement them [CPLEX, XPRESS].

Defining input for the resulting linear program may be difficult. A solution algorithm is developed in 2.6 for a third equivalent formulation, as a dynamic problem that makes the structure of the problem’s data easier to manage (see [1]).

Results in 2.7 and 2.8 are used to reduce time calculation, even if this is not a critical issue for the applications.

In section 3 a scenario analysis approach is developed, following the the ideas of [4].

Assumption in 3.2 is analogous to the one in 2.2. Now inaccuracy can originate in changes of situation between time steps of the same scenario.

In 3.4 we present again a formulation as a 0-1 non linear program. The content of 3.5 is the equivalent linear version.

In 3.6 there is an algorithm based on a dynamic (now stochastic) program formulation.

Finally, section 4 shows some examples and applications.

2. Deterministic investment problem

In what follows we are going to describe a problem that is stochastic, that is going to be solved in two stages. Even though the problem is stochastic, this first approach is going to consider a deterministic second stage problem.

At the first stage, a power system operation stochastic problem is solved for a family of sub-sets of a set of possible projects. For each operation problem, an estimation of the expected operation costs is obtained for each time step. In the operation problem, the equipment remains fixed for the time period T that is analyzed.

At this stage an operation problem must be solved for each combination of projects that is going to be considered as an alternative in the second stage. Nevertheless, this kind of operation problems is standard, and can be solved efficiently at a low cost.

As an example, stochastic dynamic programming is frequently used in hydrothermal systems to define the optimal use of water, given the level of the existing dams and a qualification of the hydrological situation (wet, medium or dry for example). Expected costs can be obtained for each time step as a result of the optimization. Sometimes the optimization algorithm requires important simplifying assumptions. In such cases, better estimates of the expected costs for each time step can be got from a more accurate simulation of scenarios, using the value of water that was calculated in the optimization algorithm. Even in that case, the operation problem for an alternative in a period of 10 years can be calculated in a few minutes, using a standard PC.

In 2.1 we give the basic definitions for the operation problem.

In 2.2 I present an assumption that is needed for the validity of the dynamic programming algorithm, and some characteristics of a power system that are suitable in order to be able to accept it.

The monotony result of 2.3 is used afterwards to reduce the volume of calculation.

2.4 is the first formulation of the second stage deterministic problem. It is an integer variables problem, that is not linear. In 2.5 there is an equivalent linear formulation of the deterministic problem.

Finally we get to the formulation in 2.6, and to an algorithm to solve the deterministic problem.

2.1 *Preliminary definitions and results*

Definition 1.
A project **p** is a pair **(f_p, co_p)** composed by a real number $\mathbf{f_p}$ and a vector $\mathbf{co_p}$. The set of all considered projects will be called **P**.

f_p is the fixed cost of the project corresponding to each time step (usually time steps are years in this framework). It includes:

- The amount of investment that has to be paid in a time step, a quantity such that its repeated addition for each time step of the project life results in the investment value, provided a constant discount rate **i** is used.
- Fixed operation and maintenance costs for each time step. Actually these costs increase with time, when projects get older. To overcome this problem, projects are going to be analysed for a period of **T** years, such that operation and maintenance costs begin increasing after **T** time steps. In addition, fixed operation and maintenance costs are much less than investments, and their variation is reduced related to their value. So it is not inaccurate to suppose them constant.
- $\mathbf{co_p}$ is a vector in a finite dimensional vector space. The components of $\mathbf{co_p}$ contain the project information needed for its adequate modelling in the operation problem. For example, if the project is a thermal plant, $\mathbf{co_p}$ will contain its installed power, availability factor, variable cost, annual time of needed preventive maintenance, etc.

T must be less than the life period of any project. As a consequence, no project can be installed and taken off before time **T**. This is a natural assumption when thinking of real investments as generation plants, due to their long life-time and their (almost) complete irreversibility. In the case of power contracts some care must be taken, and perhaps a shorter period **T** will have to be considered.

$\mathbf{co_p}$ will not be explicitly present in the formulation of the investment problem. Its values are going to be used in the operation problem to calculate the expected value of the variable cost of the system when project **p** is included in an alternative, as it will be defined later.

Definition 2.
Among the family of subsets of **P** we define a sub-family **A** of alternatives. We will use capital letters for families of alternatives, and low case letters for alternatives themselves. Each alternative is then, a subset of **P**. ***A*** can be the family of all the subsets of **P**.

Definition 3.
A finite ordered set $E = \{A_i, 1 \le i \le I\}$ of alternatives will be called an expansion if the following condition were satisfied:
$\forall (i < j),\ A_i \subseteq A_j$.
Let $tr : \{1, T\} \to A$ be a function on the family of alternatives **A**, such that $\{tr(t), 1 \le t \le T\}$ is an expansion. Then tr will be called a path.

In what follows we will only consider expansions as feasible solutions of the second stage problem. In some problems, additional constraints can reduce even more the number of alternatives. For example, suppose that k of the projects are identical; then these projects generate only one relevant alternative once the number of identical projects is fixed, with a total of k alternatives. An arbitrary set of k elements has 2^p subsets.

Definition 4.
The fixed cost of an alternative **a** is $f_a = \sum_{p \in a} f_p$ where the sum is over all the projects that belong to **a**.

Definition 5.
The variable cost of an alternative **a** is a function $v_a : \{1, T\} \to R$. This function represents the variable production cost of the whole system for each time step, when projects of alternative **a** and only them are installed. If we use a stochastic operation model for the system, $\mathbf{v_a}$ is an estimation of the expected cost.

Definition 6.
The total cost of an alternative **a** is a function $c_a : \{1, T\} \to R$ defined as: $c_a(t) = f_a + v_a(t)$.

2.2 A basic assumption

We suppose that the expected value of the system variable costs are known for an alternative **a** once we have the $\mathbf{co_p}$ vector of each project **p** belonging to **a**.

This is not strictly true when dynamical effects are present. For example, suppose the system has important reservoirs, and variable costs are estimated running an operation model of the system including the considered alternative for all the years of the study period. As a result of the optimisation operation problem, a set of Bellman values depending on the level of the dams will be calculated. Expected operation costs $\{c_a(t), 1 \le t \le T\}$ for each time step are calculated from them. These values correspond to the installation of alternative **a** alone for all the years.

Using the above mentioned operation costs, the algorithm presented here will select optimal alternatives for each time step, giving an optimal path **tr** as a result. Let $\mathbf{t_0}$ be a time step such that $tr(t_0) = a, tr(t_0 + 1) = b \supset a$. Running the operation model with the optimal path **tr** will get variable costs $\{c_{tr}(t), 1 \le t \le T\}$. Even though $tr(t_0) = a$ you can expect to have $c_{tr}(t_0) \le c_a(t_0)$, because the optimal path will “see” the additional equipment in alternative **b** for time step $\mathbf{t_0+1}$.

That kind of inaccuracy can be reduced choosing the time step long enough. For example, if reservoirs are weekly and we use a time step of one year for the power expansion model, dynamical effects could be neglected.

After a selection of the time step, the algorithm gives an optimal path **tr**. This path can be used to check the accuracy of operation costs. If results are not acceptable, a greater time step could be used.

The effect of future equipment increases with its size, and the size of the needed equipment in next time step depends on demand growth in each time step. Except with very high growing in demand, changes in future equipment are not drastic in the immediate future, and the effect of greater changes in the far future are less influent.

Using a dynamic programming operation model in a power system with monthly reservoirs, we have observed differences of less than 5% in expected operation costs, using a 1 week time step for the operation model, and a 1 year time step in the investment problem.

For the above reasons, I think that a quite accurate time step can be selected in many real life systems.

2.3 A monotony result

The typical investment problem consists in spending money to get some advantage in supplying the demand of the system. Normally, the improvement in equipment results in a reduction of costs. Let **a** and **b** be available alternatives, then: $f_a < f_b \Rightarrow v_a(t) > v_b(t), \quad \forall t \in \{1, T\}$

Even more, if we have accepted a greater investment, we expect that the system will be better equipped. The increment in costs due to an increase in demand (marginal cost) would have to be less than the increment corresponding to the same demand evolution with lower investment costs. In standard situations demand increases with time, so we can have the following continuous version for that condition: $(MC) \quad f_a < f_b \Rightarrow v'_a(t) > v'_b(t), \quad \forall t \in (0, T)$

For the problem presented bellow, the preceding monotony condition is satisfied. Suppose the operation model of a power system has the following characteristics:

- Demand is represented in blocks given by a power level and duration.
- The system has only thermal power plants.
- Dynamical effects of thermal power plants can be neglected.
- Scenarios (on demand and proportional unit cost of the plants) are known at the beginning of each time step, before the operation is done.

This model has the following formulation:

$$
(P)\left\{\begin{array}{l}
min\sum_{s=1}^{S} p(s)*\left(\sum_{t=1}^{T}\sum_{ib=1}^{B}\sum_{ith=1}^{Nth} c_{ith}(s)*P(t,ib,s,ith)*dur(t,ib)\right)\\
s.t\left\{\begin{array}{l}
\sum_{ith=1}^{Nth} P(t,ib,s,ith)*dur(t,ib)=dem(t,ib,s), \quad 1\le t\le T, 1\le ib\le B, 1\le s\le S \quad (1)\\
P(t,ib,s,ith)\le P^{\sup}(ith), \quad 1\le t\le T, 1\le ib\le B, 1\le s\le S, 1\le ith\le Nth \quad (2)\\
P(t,ib,s,ith)\ge 0, \quad 1\le t\le T, 1\le ib\le B, 1\le s\le S, 1\le ith\le Nth \quad (2)
\end{array}\right.
\end{array}\right. .
$$

where:
p(s) is the probability of scenario **s**.
c_{ith}(s) is the variable cost of thermal plant **ith** ($/MWh) in scenario **s**.
P(t,ib,s,ith) is power (MW) of thermal plant **ith** that has been dispatched in block **ib** of time step **t**, for scenario **s**.
dur(t,ib) is the duration (hours) of block **ib** of time step **t**
dem(t,ib,s) is the power demand (MW) corresponding to block **ib** of time step **t**, for scenario **s**.
P^{sup}(ith) is the installed capacity (MW) in thermal plant ith

Installed capacity (the goal of the investment problem) works as an upper bound in the operation problem. Correspondingly, expanding capacity (that correspond to an increase in fixed costs) accounts for relaxing the operation problem.

In this case it is possible to solve separately a dispatching problem for each demand block **ib**, of each time step **t** and each scenario **s**.

$$
(P_{t,ib,s})\left\{\begin{array}{l}
min\ p(s)*\left(\sum_{ith=1}^{Nth} c_{ith}(s)*P(t,ib,s,ith)*dur(t,ib)\right)\\
s.t.\left\{\begin{array}{l}
\sum_{ith=1}^{Nth} P(t,ib,s,ith)*dur(t,ib)=dem(t,ib,s), \quad (1)\\
P(t,ib,s,ith)\le P^{\sup}(ith), \quad 1\le ith\le Nth \quad (2)\\
P(t,ib,s,ith)\ge 0, \quad 1\le ith\le Nth \quad (2)
\end{array}\right.
\end{array}\right.
$$

The Lagrangean of ($P_{t,ib,s}$) is:

$$L(P,\lambda,\mu,\nu) = p(s)*dur(t,ib)*\left(\sum_{ith=1}^{Nth} c_{ith}(s)*P(t,ib,s,ith)\right)+$$
$$\lambda*\left(-\sum_{ith=1}^{Nth} P(t,ib,s,ith)*dur(t,ib)+dem(t,ib,s)\right)+\sum_{ith=1}^{Nth}\mu(ith)*\left(P(t,ib,s,ith)-P^{\sup}(ith)\right)-$$
$$-\sum_{ith=1}^{Nth}\nu(ith)*P(t,ib,s,ith)$$

The (KKT) conditions are:

$$(1)\begin{cases}\left.\begin{array}{l}P(t,ib,s,ith)\le P^{\sup}(ith)\\ P(t,ib,s,ith)\ge 0\end{array}\right\}1\le ith\le Nth\\ -\sum_{ith=1}^{Nth} P(t,ib,s,ith)*dur(t,ib)+dem(t,ib,s)=0\end{cases}$$ (primal feasibility conditions)

$$(2)\begin{cases}\mu(ith)\ge 0\\ \nu(ith)\ge 0\end{cases} 1\le ith\le Nth$$ (dual feasibility conditions)

$$(3)\begin{cases}\lambda*\left(-\sum_{ith=1}^{Nth} P(t,ib,s,ith)*dur(t,ib)+dem(t,ib,s)\right)=0\\ \left.\begin{array}{l}\mu(ith)*\left(P(t,ib,s,ith)-P^{\sup}(ith)\right)=0\\ \nu(ith)*P(t,ib,s,ith)=0\end{array}\right\}1\le ith\le Nth\end{cases}$$ (complementary slackness)

$$(4)\{(p(s)*c_{ith}(s)-\lambda)*dur(t,ib)+\mu(ith)-\nu(ith)=0,\ 1\le ith\le Nth$$ (hidden constraints)

For each plant ith, three cases can be distinguished:

$$Case\ 1\quad p(s)*c_{ith}(s)<\lambda\Rightarrow(p(s)*c_{ith}(s)-\lambda)*dur(t,ib)<0\underset{(4)}{\Rightarrow}\mu(ith)-\nu(ith)>0\underset{\nu(ith)\ge 0}{\Rightarrow}$$
$$\mu(ith)>0\underset{(3)}{\Rightarrow}P(t,ib,s,ith)=P^{\sup}(ith)>0\underset{(3)}{\Rightarrow}\nu(ith)=0\underset{(4)}{\Rightarrow}\frac{\mu(ith)}{dur(t,ib)}=\lambda-p(s)*c(ith)>0$$

So that the plant has a complete dispatch in this case, and increasing P^{sup}(ith) results in a reduction of the optimal value.

$$Case\ 2\quad p(s)*c_{ith}(s)>\lambda\Rightarrow(p(s)*c_{ith}(s)-\lambda)*dur(t,ib)>0\underset{(4)}{\Rightarrow}\nu(ith)-\mu(ith)>0\underset{\mu(ith)\ge 0}{\Rightarrow}$$
$$\nu(ith)>0\underset{(3)}{\Rightarrow}P(t,ib,s,ith)=0\underset{(3)}{\Rightarrow}\mu(ith)=0\underset{(4)}{\Rightarrow}\frac{\nu(ith)}{dur(t,ib)}=p(s)*c(ith)-\lambda>0$$

In this case the plant ith is not dispatched at all.

$$Case\ 3\quad p(s)*c_{ith}(s)=\lambda\Rightarrow(p(s)*c_{ith}(s)-\lambda)*dur(t,ib)=0\underset{(4)}{\Rightarrow}\nu(ith)=\mu(ith).$$
$$\nu(ith)>0\underset{(3)}{\Rightarrow}P(t,ib,s,ith)=0\underset{(3)}{\Rightarrow}\mu(ith)=0\underset{(4)}{\Rightarrow}\nu(ith)\ne\mu(ith)\Rightarrow\nu(ith)=\mu(ith)=0$$
$$¡\lambda=p(s)*c_{ith}(s)>0\underset{(3)}{\Rightarrow}dem(t,ib,s)=\sum_{ith=1}^{Nth}P(t,ib,s,ith).$$

So that demand is satisfied if some plant is in Case 3.

In order to satisfy the above conditions, these problems are solved adding power of plants in increasing order of $c_{ith}(s)$ until $P^{sup}(ith)$ is reached (thermal plant with complete dispatch) or demand is satisfied (marginal plant). Plants with $c_{ith}(s)$ greater than the one of the marginal plant are not dispatched for the current (t,ib,s).

Let **cm(t,ib,s)** be the variable cost of the marginal plant, **im** its thermal plant index. Then we have:

$$\lambda(t,ib,s) = p(s) * cm(t,ib,s)$$

$$\mu(t,ib,s,ith) = \begin{cases} p(s) * (cm(t,ib,s) - c_{ith}(s)), & c_{ith}(s) \leq cm(t,ib,s) \\ 0, & c_{ith}(s) > cm(t,ib,s) \end{cases}$$

Let ΔP be a differential increase in $P^{sup}(ith)$.
The corresponding unit reduction of operation costs is then

$$\sum_{t=1}^{T} \sum_{ib=1}^{B} \sum_{s=1}^{S} p(s) * (cm(t,ib,s) - c_{ith}(s))^{+} = E\left[\sum_{t=1}^{T} \sum_{ib=1}^{B} (cm(t,ib,s) - c_{ith}(s))^{+}\right] \geq 0$$

This shows that the expected value of operation costs decreases with the increase in equipment.

Consider now ΔP as a finite increase. To see the effect on marginal costs, we distinguish the same three cases as above.

Case 1 $p(s) * c_{ith}(s) < \lambda$ Plant ith has a complete dispatch.

- If $\Delta P \leq P(t,ib,s,im)$ then the marginal plant reduces its dispatch by ΔP and marginal cost remains unchanged.

Let us call MP to the difference between the demand dem(t,ib,s) and the power dispatched until plant ith (including ith).

- If $P(t,ib,s,im) < \Delta P \leq MP$ then the marginal plant have no dispatch and marginal cost is reduced to a value between the initial marginal cost and $c_{ith}(s)$.
- If $MP < \Delta P$ then all the plants with variable cost greater than $c_{ith}(s)$ have no dispatch and marginal cost is reduced to the value $c_{ith}(s)$.

So marginal cost does not increase in Case 1.

Case 2 $p(s) * c_{ith}(s) > \lambda$ Plant ith had no dispatch before increasing its maximum power, and it will still having no dispatch afterwards. Marginal cost does not change in this case.

Case 3 $p(s) * c_{ith}(s) = \lambda$ In this case **ith=im** Plant ith has a partial dispatch before increasing its maximum power, and it will still having the same dispatch afterwards. Marginal cost does not change in this case.

We conclude that for each time step **t**, demand block **ib** and scenario **s**, marginal cost does not increase. Then the same is true for expected values or average marginal costs, regardless of the type of weights used.

Each of these problems then satisfies the (MC).

More complex models represent dynamical effects of thermal plants, associated to the start up process. If the power system has hydro plants with important dams, then dynamic links between time steps become important. I do not have a demonstration for the more complex cases.

For the proof of the following Proposition we accept a continuous version of (MC). In this framework, we suppose the variable system cost function ν is of class C^1 for each alternative **a** in **A**.

Proposition 1.
Let us suppose that (MC) is satisfied for the pair of alternatives **a** and **b** in **A**.
Suppose in addition that $f_a < f_b, t^* \in (0,T)$. Then:
a) $c_a(t^*) > c_b(t^*) \Rightarrow c_a(t) > c_b(t), \forall t > t^*$
b) $c_a(t^*) < c_b(t^*) \Rightarrow c_a(t) < c_b(t), \forall t < t^*$

Proof
As $\mathbf{f}_a$ is fixed and ν_a is of class $\mathbf{C}^1$ the following is verified for each alternative **a**:

$$c_a(t) = c_a(t^*) + \int_{t^*}^{t} \nu_a'(u)du, \quad \forall t > t^*$$

$$f_a < f_b \Rightarrow \nu_b'(t) < \nu_a'(t), \forall t \in (0,T) \Rightarrow$$

$$c_b(t) = c_b(t^*) + \int_{t^*}^{t} \nu_b'(u)du < c_a(t^*) + \int_{t^*}^{t} \nu_a'(u)du = c_a(t), \forall t > t^*$$

This proves (a). The proof of (b) is similar.

2.4 Formulation of the problem

If **nP** is the number of possible projects and there is no additional constraints, then the number **nA** of alternatives will be $\mathbf{2^{nP}}$.

Additional constraints on the alternatives can reduce its number. As an example, if coal is not yet used for power generation, and **p1** and **p2** are two possible first projects with this fuel, they will include infrastructure costs. Each of the above mentioned projects, considered as a second project (after doing the other) will be cheaper than the same project, considered as a first one; let us consider them as different projects **p21** (project 2 after doing project 1) and **p12** (project 1 after doing project 2). Then **p21** can not belong to an alternative if **p1** does not belong to it.

Constraints that depend on time (like financial constraints) will not be considered. Some of the proof use that constraints are explicitly dependent of the time elapsed.

Alternatives will be referenced with index **a**. A great number of alternatives **nA** can make unpractical the algorithms studied in this work. In such a situation, some sets of projects could be considered as a single one, especially if they are of the same type. For example, if 20 MW installed capacity gas turbines are to be installed in a system where demand increases by 500 MW each year, possibly identical gas turbine projects could be considered in groups of 5.

We will suppose $\nu_A(t)$ is a good approximation for the expected value of the variable cost at time step **t** for the optimal path **tr**. This approximation can have some inaccuracy (see 2.2 for details). Even if that assumption is not valid, the solution to the following problem will give a good initial feasible solution for the investment problem. Such an initial solution can be useful for more exact algorithms, like branch and bound methods.

The deterministic investment problem has the following formulation:

$$(P_1)\begin{cases} min\displaystyle\sum_{t=1}^{T}\sum_{a=1}^{nA} x_a(t)*\frac{c_a(t)}{(1+i)^t} \\ s.t.\begin{cases} \displaystyle\sum_{a=1}^{nA} x_a(t)=1 \quad 1\le t\le T \quad (1.t) \\ \displaystyle\sum_{a=1}^{nA} x_a(t)*\sum_{b\in I_a} x_b(t-1)=1 \quad 2\le t\le T \quad (2.t) \\ x_a(t)\in\{0,1\} \quad 1\le a\le nA, 1\le t\le T \quad (3.t) \end{cases} \\ where \quad I_a=\{b: transition\ b\to a\ is\ allowed\}\subseteq\{b: b\subseteq a\} \end{cases}$$

Decision variables $\mathbf{x_a(t)}$ are equal to one only if alternative **a** is chosen in time step **t**. The objective function represent the present value of total costs, where $\mathbf{c_a(t)}$ includes fixed and operation costs and **i** is the discount rate.

Constraint **(1.t)** establish that exactly one alternative is chosen at each time step.

In constraints **(2.t)** we represent the irreversibility of investments. We know by **(1.t)** that given the time step, there is only one alternative **a** such that $\mathbf{x_a(t)}$ is one. In order to satisfy **(2.t)**, for the alternative of index **a**, exactly one alternative belonging to $\mathbf{I_a}$ must be chosen at time step **t-1**, where $\mathbf{I_a}$ is the set of alternatives from which transition to alternative **a** is possible.

2.5 Linear version

Problem (P_1) is not linear. In order to have a linear formulation; we will define new variables $z_a(t)=x_a(t)*\sum_{b\in I_a} x_b(t-1), \quad 2\le t\le T, \quad 1\le a\le A$. These variables are going to be defined linearly using the following Proposition.

Proposition 2.

We consider $\begin{cases} X, Y \in \{0,1\}, \\ Z = X * Y, \\ W(X,Y) = \{w \in \{0,1\}, w \leq X, w \leq Y, w \geq X + Y - 1\} \end{cases}$

Then $W(X,Y) = \{X * Y\}$.

Proof.

Suppose first that X=0.

$$X = 0 \Rightarrow \begin{cases} Z = 0 \\ \left.\begin{array}{l} X + Y - 1 \leq Y - 1 \leq 0 \\ 0 \leq X,\ 0 \leq Y \end{array}\right\} \Rightarrow 0 \in W(0,Y) \\ 1 > X \Rightarrow 1 \notin W(0,Y) \end{cases}$$

The same proof is valid for Y=0. Now suppose X=Y=1.

$$X = Y = 1 \Rightarrow \begin{cases} Z = 1 \\ \left.\begin{array}{l} X + Y - 1 = 1 \\ 1 \leq X,\ 1 \leq Y \end{array}\right\} \Rightarrow 1 \in W(1,1) \\ 0 < X + Y - 1 = 1 \Rightarrow 0 \notin W(1,1) \end{cases}$$

And the result follows.

As $x_a(t), \sum_{b \in I_a} x_b(t-1) \in \{0,1\}$, the linear problem ($P_2$) is equivalent to ($P_1$).

$$(P_2)\begin{cases} min \displaystyle\sum_{t=1}^{T} \sum_{a=1}^{nA} x_a(t) * \frac{c_a(t)}{(1+i)^t} \\ s.t. \begin{cases} \displaystyle\sum_{a=1}^{nA} x_a(t) = 1 \quad 1 \leq t \leq T \quad (1.t) \\ \displaystyle\sum_{a=1}^{nA} z_a(t) = 1 \quad 2 \leq t \leq T \quad (2.t) \\ z_a(t) \leq x_a(t) \quad 1 \leq a \leq nA \quad 2 \leq t \leq T \quad (3.a.t) \\ z_a(t) \leq \displaystyle\sum_{b \in I_a} x_b(t-1) \quad 1 \leq a \leq nA \quad 2 \leq t \leq T \quad (4.a.t) \\ x_a(t) + \displaystyle\sum_{b \in I_a} x_b(t-1) - z_a(t) \leq 1 \quad 1 \leq a \leq nA \quad 2 \leq t \leq T \quad (5.a.t) \\ x_a(t), z_a(t) \in \{0,1\} \quad 1 \leq a \leq nA, \forall t \quad (6.a.t) \end{cases} \end{cases}$$

2.6 Dynamic programming type recurrence

In what follows we are going to suppose (MC) is verified, so that the monotony result of 2.3 is valid.

Multi-stage stochastic programs can be described in terms of dynamic programming. A standard formulation can be found in [1]. We present now a first equivalent formulation for problem (P_1).

Let us consider the optimal value **fopt(a,t)** corresponding to time steps from **t** on, being given that alternative **a** holds during time step **t**. An expression for that function is:

$$fopt(a,t) = c_a(t) + vopt(a,t)$$

$$where: \quad vopt(a,t) = min \sum_{u=t+1}^{T} \sum_{d \in J_a} \frac{x_d(u) * c_d(u)}{(1+i)^{u-t}}$$

$$s.t. \begin{cases} \sum_{d \in J_a} x_d(u) = 1, & t+1 \le u \le T \\ \sum_{d \in J_a} x_d(u) * \sum_{b \in J_d} x_b(u+1) = 1, & t+1 \le u \le T-1 \end{cases}$$

$$J_a = \{d : transition\ a \to d\ is\ allowed\} \subseteq \{d : a \subseteq d\}$$

$$J_d = \{b : transition\ d \to b\ is\ allowed\} \subseteq \{b : d \subseteq b\}$$

fopt can be evaluated in a dynamical programming framework. It will be defined by the following recursive formulas:

$$fopt(a,t) = c_a(t) + \frac{\min_{d \in J_a} fopt(d, t+1)}{(1+i)}, \quad 1 \le t \le T-1, \quad 1 \le a \le nA$$

$$fopt(a,T) = c_a(T), \quad 1 \le a \le nA$$

The following dynamical programming algorithm implements the method above. To store some useful information concerning the optimal path, we will define the integer function **trajec**.

Definition 7.

We define the function $trajec : \{1, A\} \times \{1, T\} \to \{1, A\}$ such that:

- If **t<T**, then **trajec(a,t)** points to the following alternative in the optimal path beginning at feasible alternative **a** at time step **t**
- If **t=T**, **trayec(a,T)=a**

We can now define the following

Algorithm (deterministic)

1) Initialise

$fopt(a,T) = c_a(T)$

$trajec(a,T) = a, \quad 1 \le a \le nA$

2) Main process

$$t = (T-1:-1:1), a = (1, nA) \begin{cases} Calculate \\ J_a = \{d : transition\ a \to d\ is\ allowed\} \\ m = \min\limits_{d \in J_a}\{fopt(d, t+1)\} \\ im = \arg\min\limits_{d \in J_a}\{fopt(d, t+1)\} \\ fopt(a,t) = c_a(t) + \dfrac{m}{(1+i)} \\ trajec(a,t) = im \end{cases}$$

At the end of main process, **fopt(a,1)** contains the optimal values for the problem beginning with each alternative in time step 1. Once the alternative that holds in the first time step is determined, **trajec** contains enough information to construct the optimal trajectory.

2.7 Using the monotony result

The aim is to reduce the number of transitions to be tested. The set $\mathbf{J_a}$ contains alternatives including **a** and satisfying other constraints, and it does not depend on time step **t**.

We suppose transitivity:

$$\left.\begin{matrix} a \to b\ allowed \\ b \to c\ allowed \end{matrix}\right\} \Rightarrow a \to c\ allowed$$

The following proposition permits to reduce these sets using information produced in the main algorithm.

Proposition 3.

Let **t** be a time step, alternatives **a** and **b**, a previous time step **u** and a path **tr** such that

$$\begin{cases} 1 \le u < t \le T\ (1) \\ b \in J_a\ (2) \\ c_a(t) \le c_b(t)\ (3) \\ tr(u) = a\ (4) \\ tr(u+1) = b\ (5) \end{cases}$$

Then there is a path **btr** with a cost less or equal than the cost of **tr** and such that $\begin{matrix} btr(u) = a \\ btr(u+1) = a \end{matrix}$.

Using this result repeatedly, it can be concluded that there is a path without any transition from alternative **a** to alternative **b** in a time step **u** with **u<t** and with cost less or equal to that of **tr**. It is then possible to exclude the possibility of a transition from alternative **a** to alternative **b** for the rest of the algorithm.

Proof.
Let us define an integer function **btr** from **tr** in the following way:
$btr : \{1, T\} \to A$

$$btr(s) = \begin{cases} a, \; s = u+1 \\ tr(s)\; otherwise \end{cases}$$

btr is equal to **tr** except for **t=u+1**, where **tr** has a transition from **a** to **b** and **btr** remains in **a**.

$$\left. tr(u+1) = b \Rightarrow \quad \begin{matrix} tr(u+2) \in J_b \\ b \in J_a \end{matrix} \right\} \Rightarrow btr(u+2) = tr(u+2) \in J_a$$

This shows that the transition from $btr(u+1) = a$ to $btr(u+2) = tr(u+2)$ is allowed, so that **btr** is a path.

$b \in J_a \Rightarrow a \subseteq b \Rightarrow f_a \le f_b$
Using (3) and the monotony result, we have
$\forall u \le t, c_a(u) \le c_b(u) \, (6)$

The total cost function of **btr** is the same as the one of **tr**, except for time step **u+1**, where $\mathbf{c_a(u+1)} \le \mathbf{c_b(u+1)}$ (see (6)). So the path **btr** has total cost less or equal than **tr**, and it has avoided the transition from alternative **a** to alternative **b** in time step **u<t**.

Let us consider time step **t** in the backward iteration of the dynamic programming algorithm of 2.6, and suppose alternatives **a** and **b** are as in Proposition 3. We can then eliminate **b** from the set $\mathbf{J_a}$ for iterations corresponding to time steps **u<t**.

Condition **(2)** in Proposition 3 means that alternative **b** have “too much investment for the size of the system in time step **t**”. For previous time steps, the system will be smaller, so that alternative **b** will be still too big in investment.

Last proposition has been used to reduce the number of transitions in the main algorithm.

Note: the same proof can be done for a path **tr** with

$$\left. \begin{matrix} tr(u) = c \\ tr(u+1) = b \end{matrix} \right\} 1 \le u < t \le T, \quad a \in J_c, \quad b \in J_a, \quad c_a(t) \le c_b(t)$$

In that case, we have to define $btr(s) = \begin{cases} a, & s = u+1 \\ tr(s) & otherwise \end{cases}$

The following Proposition establishes an analogous result for **u>t**.

Proposition 4.
Let **t** be a time step, alternatives **a** and **b**, a time step **u>t** and a path **tr** such that

$$\begin{cases} 1 \le t < u \le T \ (1) \\ a \in J_b \ (2) \\ c_a(t) \le c_b(t) \ (3) \\ tr(u) = b \ (4) \\ tr(u+1) \in J_a \ (5) \end{cases}$$

Then there is a path **btr** with a cost less or equal than the cost of **tr** and such that $btr(u) = a$.

Proof
Let us define an integer function **btr** from **tr** in the following way:

$$btr : \{1, T\} \to A$$

$$btr(s) = \begin{cases} a, \ s = u \\ tr(s) \ otherwise \end{cases}$$

btr is equal to **tr** except for **t=u**, where **tr** is equal to alternative **b** and **btr** is equal to alternative **a**.

$$\left.\begin{array}{l} btr(u+1) = tr(u+1) \in J_a \\ btr(u) = a \end{array}\right\}$$ This shows that the transition from $btr(u) = a$ to $btr(u+1) \in J_a$ is allowed, so that **btr** is a path.

Using the monotony result

$$\left.\begin{array}{l} a \in J_b \Rightarrow b \subseteq a \Rightarrow f_b < f_a \\ c_a(t) \le c_b(t) \\ u > t \end{array}\right\} \Rightarrow c_a(u) \le c_b(u)$$

As the only difference between **tr** and **btr** is that **tr(u)=b** and **btr(u)=a**, then **btr** is a path with a cost less or equal than the cost of **tr** and such that $btr(u) = a$.

As the main algorithm proceeds backward, the use of this result to modify the set $\mathbf{J_b}$ would require a pre-processing procedure.

Conditions **(2)** and **(3)** in Proposition 4 mean that alternative **b** have “not enough investment for the size of the system in time step **t**, so that its variable costs are too big”. For a following time step **u**, the system will be greater, so that alternative **b** will be still too small in investment for the current time step.

Condition **(5)** is necessary. Suppose a very good path for period **[u+1,T]** can be reached from alternative **b** but is not accessible from alternative **a**. In such case, **b** could be better than **a** in time step **u**.

2.8 Improvement inside a time step

The deterministic algorithm presented above has a (backward) loop in time steps and a loop in alternatives. For each time step, the process consist in analyzing transitions from each alternative. In the middle of this process, some alternatives are yet studied and others are not. Information from yet studied alternatives is useful for the remaining ones, as is shown in the following proposition.

Proposition 5.

Let us define:

$$J_a = \{b : a \to b \text{ is allowed}\} \subseteq \{b : a \subseteq b\}$$

$$I_a = \{c : c \to a \text{ is allowed}\} \subseteq \{c : c \subseteq a\}$$

$$im = \arg\min_{b \in J_a} \left(fopt(b, t+1) \right)$$

$\mathbf{J_a}$ is the set of alternatives **b** to which transition is possible from **a**.
$\mathbf{I_a}$ is the set of alternatives **c** such that transition is possible from **c** to **a**.
Then the transitions from an alternative **c** with index in $\mathbf{I_a}$ to an alternative **b** with index in $\mathbf{J_a}$ is never better than the transition from **c** to the alternative of index **im**.

Proof.

Let **tr** be a path with a transition from an alternative **c** belonging to $\mathbf{I_a}$ in time step **t**, to an alternative **b** belonging to $\mathbf{J_a}$ in time step **t+1**, that is:

$$tr : \{1, T\} \to A$$

$$tr(t) = c \in I_a$$

$$tr(t+1) = b \in J_a$$

Let us define an integer function **btr** from **tr** in the following way:

$$btr : \{1, T\} \to A$$

$$btr(u) = \begin{cases} tr(s), & u \le t \\ im, & u = t+1 \end{cases}$$

and **btr** optimal for **u≥t+2**.

$$\left.\begin{matrix} c \in I_a \\ im \in J_a \end{matrix}\right\} im \in J_c$$ This shows that the transition from **c** to **im** is allowed, so that **btr** is a path.

As fopt(im,t+1) ≤fopt(b,t+1), $\forall$ b $\in$ J_a, then btr is not worse than tr.

3. Generalisation: optimisation under uncertainty

Suppose now we can represent uncertainty using a finite number or scenarios with assigned probabilities.

3.1 Definitions

Let us define the following elements of such a setting for the stochastic problem.

Definition 8.
A scenario **scen** is a matrix of dimensions **nA*T** such that **scen(a,t)** is a possible future total cost associated with the **a** feasible alternative at time step **t**. It represent the behaviour of the system in a possible future period **[1,T]**, for each feasible alternative.

Definition 9.
The set of all considered scenarios is $SC = \{scen(.,.,sc), 1 \leq sc \leq nSC\}$, where **nSC** is the number of scenarios.

In this problem we are going to consider a finite number of scenarios. As the scenarios are themselves matrices, in what follows we are going to consider the set of scenarios as a unique matrix of dimensions **nA*(T*nSC)**.

In the following applications we shall store only a limited number of situations, from which scenarios are constructed. Concretely, each scenario will consist in a sequence of situations.

If the number of recorded situations is **nS**, then we only have to allocate memory for **nA*(T*nS)** numbers.

Definition 10.
Let **Bun** be a partition of **SC**, that is

$$Bun = \left\{ B(i), 1 \leq i \leq I, \bigcup_{i=1}^{I} B(i) = SC, i \neq j \Rightarrow B(i) \cap B(j) = \phi \right\}$$

Each set **B(i)** of **Bun** will be called a bundle of **SC**.

Definition 11.
We take $K = \{Bun(t) | Bun(1) = \{SC\}, Bun(T) = \{\{scen(.,.,sc)\}, 1 \le sc \le nSC\} 1 \le t \le T\}$.
K is a sequence of partitions of SC such that the first one has the set **SC** as its unique element and the last one is the partition of **SC** in singletons.

We will call **K** the evolution of our knowledge about the system. In this framework a bundle belonging to **Bun(t)** will be called **B(i,t)**.

Scenarios belonging to a set **B(i,t)** represent the ones that can not be distinguished at time step **t**, given that **K** is the evolution of our knowledge about the system.

The first condition means all scenarios are possible at the beginning, so that we are not able to discard any of them using information available at time step 1.

The second condition means we know the individual scenario that took place when we are in time step **T**. If that condition is not fulfilled, each indistinguishable set of scenarios in time step **T** could have been considered as a single scenario. We are going to make the assumption that the partitions in the evolution of our knowledge **K** are refined when **t** increases. This means we do not forget the passed history. We are using here the model of scenario analysis of [4].

Definition 12.
We will call **nb(t)** the number of bundles in **Bun(t)**. That is:
$Bun(t) = \{B(i,t), 1 \le i \le nb(t)\}$

Definition 13.
We will call **JBun$_i$(t)** the set of indices of bundles of time step **t+1** such a transition from **B(i,t)** to bundles in **JBun$_i$(t)** is allowed, that is:
$JBun_i(t) = \{j | B(j,t+1) \in Bun(t+1), P(B(j,t+1) | B(i,t)) > 0\}$

Definition 14.
The probability function of the scenarios is a function
$pr : \{1, \cdots, nSC\} \to (0,1)$
$pr(sc) \ge 0$
$\sum_{sc=1}^{nSC} pr(sc) = 1$
pr(sc) is the probability of scenario **scen(sc)**.

In a finite number of scenarios setting, **pr(s) = 0** and **pr(s) = 1** are not interesting possibilities. The first one should be eliminated. The second one implies **pr(s) = 0** for the other scenarios, so that there is only one relevant scenario. That situation corresponds to the deterministic case, which we have yet studied. These arguments are not still valid for an infinite number of scenarios.

3.2 Basic assumption for the stochastic problem

We make the basic assumption that operation costs are well defined for an alternative **a**, a time step **t** and a situation **s**, regardless of the future alternatives and situations that will be present in each feasible strategy and each scenario **sc**, as long as that situation **s** belongs to **sc** in time step **t**.

This assumption is of the same nature as the one in the deterministic approach.

Here we have a new source of inaccuracy. The estimation of operation costs is made in a fixed situation. This is accurate only for scenarios that have the same situations for time step **t** on. In other cases there is an error associated to the estimation. Special care must be taken when there is a great conditional probability of changing between very different situations in a time step, and specially if such a framework could conduct easily to a change in the chosen alternative in the same time step.

3.3 Conditional Transition probabilities between bundles

Definition 15.
Suppose we have an evolution **K** of our knowledge about the system and a probability function **pr** of the scenarios in **SC**.
We consider $B(i,t) \in Bun(t), B(j,t+1) \in Bun(t+1)$. We will use the conditional transition probability definition $P(B(j,t+1)|B(i,t)) = \frac{P(B(j,t+1) \cap B(i,t))}{P(B(i,t))}$ as usual in probability theory. We will call it the conditional transition probability between bundle **i** in time step **t** and bundle **j** in time step **t+1**.

Proposition 6.
Knowing the probability function **pr** of the scenarios is equivalent to knowing the conditional transition probabilities between bundles.

Proof
Suppose first that we know the probability function **pr** of the scenarios. Then by definition

$$P(B(j,t+1)|B(i,t)) = \frac{\sum_{sc \in E} pr(sc)}{\sum_{sc \in D} pr(sc)}$$

$$E = \{sc : scen(.,.,sc) \in B(j,t+1) \cap B(i,t)\}$$

$$D = \{sc : scen(.,.,sc) \in B(i,t)\}$$

Reciprocally, suppose now we know the conditional transition probabilities between bundles. We shall calculate the individual probabilities of scenarios using the following recursive formulas:

For **t=1** $Bun(1) = \{B(1,1)\} = \{SC\} \Rightarrow P(B(1,1)) = P(C) = 1$

Suppose now that we know the bundle probabilities $P(B(i,t)), 1 \leq i \leq I(t)$ for a time step **t**, and the conditional probabilities

$P(B(j,t+1) | B(i,t)), 1 \leq j \leq I(t+1), 1 \leq i \leq I(t)$.

Then $P(B(j,t+1)), 1 \leq j \leq I(t+1)$ can be determined:

$$P(B(j,t+1)) = \sum_{i=1}^{I(t)} [P(B(j,t+1) \cap B(i,t))] = P(B(i,t)) * \sum_{i=1}^{I(t)} P(B(j,t+1) | B(i,t))$$

Using this formula repeatedly it is possible to calculate the probability of bundles for time step **T**. But then

$B(i,T) = \{scen(i)\}, 1 \leq i \leq I(T) = nSC \Rightarrow$

$P(B(i,T)) = pr(i)$

This shows that transition probabilities have all the probability information of **K**. For each time step, we can record these conditional probabilities in a matrix.

Definition 16.

Let **t** be a time step. The transition probability matrix $\mathbf{prtr_t}$ corresponding to time step **t** is the matrix with generic element

$$prtr_t(i,j) = P(B(j,t+1) | B(i,t))$$
$$1 \leq i \leq I(t), 1 \leq j \leq I(t+1)$$

This matrix is of dimensions $nb(t) * nb(t+1)$where **nb(t)** is the number of bundles in **Bun(t)**. It will be denoted simply **prtr** if there is no ambiguity.

This rectangular matrix works in the stochastic algorithm like the square transition probability matrix between states of a Markov chain in a stochastic dynamic programming algorithm.

3.4 The investment problem under uncertainty

As in the deterministic case, we begin with the setting of the problem for the stochastic case.

We will assume that we have an estimation of the expected operation costs for each alternative **a** and each time step **t**, given that we are in bundle **B(i,t)**. As scenarios in **B(i,t)** can not be distinguished, they must all have the same sequence of situations for **1≤u≤t**. So it is enough to know the expected operation costs for each alternative **a**, each time step **t** and each situation **s**.

$$
(S_1)\begin{cases}
min\sum_{t=1}^{T-1}\frac{1}{(1+i)^t}\sum_{i\in IBun(t)}p(B(i,t))*\sum_{a=1}^{nA}x(a,t,i)*\left(f_a+E(\nu_a(t+1)|B(i,t))\right)\\
s.t.\begin{cases}
\sum_{a=1}^{nA}x(a,t,i)=1,\quad 1\le t\le T-1,\quad i\in IBun(t),\quad (1.t.i)\\
\sum_{a=1}^{nA}x(a,t,i)*\left(\sum_{b\in J_a}x(b,t+1,j)\right)=1, 1\le t\le T-1, i\in IBun(t),\ j\in JBun_i(t), (2.t.i.j)\\
x(a,t,i)\in\{0,1\},\quad 1\le a\le nA,\quad 1\le t\le T,\quad i\in IBun(t),\quad (3.a.t.i)
\end{cases}
\end{cases}
$$

$$
\begin{aligned}
&where\ IBun(t)=\{i|B(i,t)\in Bun(t), 1\le i\le nb(t)\}\\
&JBun_i(t)=\{j|B(j,t+1)\in Bun(t+1), P(B(j,t+1)|B(i,t))>0\}\\
&J_a=\{b: transition\ a\to b\ is\ allowed\}\subseteq\{b:a\subseteq b\}
\end{aligned}
$$

In the stochastic problem, **x(a,t,i)=1** if in time step **t** and if bundle **B(i,t)** occur at time step **t**, you decide to install alternative **a** from time step **t+1**.

Non-anticipation constraints imply that there is only one decision variable **x(a,t,i)** for all the scenarios belonging to bundle **B(i,t)**.

The objective function is an expression of the expected value of the total costs corresponding to a valid strategy. For the expected value we have the expression $E(\nu_a(t+1)|B(i,t))=\sum_{j\in JBun_i(t)}P(B(j,t+1)|B(i,t))\nu_a(t+1,j)$

where $\nu_a(t+1,j)$ is the variable cost corresponding to time step **t+1** if alternative **a** is installed and bundle **B(j,t+1)** occurs.

Constraints **(1.t.i)** establish that exactly one alternative is chosen in every feasible strategy for each bundle **B(i,t)** of scenarios in each time step **t**.

Constraints **(2.t.i.j)** establish that alternative to be chosen in time step **t+1** must belong to the set $\mathbf{J_a}$ of alternatives accessible from **a**.

Constraints **(3.a.t.i)** define **x(a,t,i)** as logical variables.

3.5 Linear version

Problem (S_1) is not linear. In order to have a linear formulation; we will define new variables

$$z(a,t,i,j) = x(a,t,i) * \left(\sum_{b \in J_a} x(b,t+1,j) \right), 1 \le a \le nA, 1 \le t \le T-1, i \in IBun(t), \ j \in JBun_i(t)$$

These variables are going to be defined linearly as done in the deterministic problem. The resulting equivalent linear problem is

$$(S_L) \begin{cases} min \sum_{t=1}^{T-1} \frac{1}{(1+i)^t} \sum_{i \in IBun(t)} p(B(i,t)) * \sum_{a=1}^{nA} x(a,t,i) * \left(f_a + E(v_a(t+1) | B(i,t)) \right) \\ s.t. \begin{cases} \sum_{a=1}^{nA} x(a,t,i) = 1, \quad 1 \le t \le T-1, \quad i \in IBun(t), \quad (1.t.i) \\ \sum_{a=1}^{nA} z(a,t,i,j) = 1, 1 \le t \le T-2, i \in IBun(t), \ j \in JBun_i(t), (2.t.i.j) \\ z(a,t,i,j) \le x(a,t,i), 1 \le a \le nA, 1 \le t \le T-2, i \in IBun(t), \ j \in JBun_i(t), (3.a.t.i.j) \\ z(a,t,i,j) \le \sum_{b \in J_a} x(b,t+1,j), 1 \le a \le nA, 1 \le t \le T-2, i \in IBun(t), \\ j \in JBun_i(t), (4.a.t.i.j) \\ x(a,t,i) + \sum_{b \in J_a} x(b,t+1,j) - z(a,t,i,j) \le 1, 1 \le a \le nA, 1 \le t \le T-2, i \in IBun(t), \\ j \in JBun_i(t), (5.a.t.i.j) \\ x(a,t,i) \in \{0,1\}, \quad 1 \le a \le nA, \quad 1 \le t \le T-1, \quad i \in IBun(t), \quad (6.a.t.i) \end{cases} \end{cases}$$

$$where \ IBun(t) = \{ i | B(i,t) \in Bun(t), 1 \le i \le nb(t) \}$$

$$JBun_i(t) = \{ j | B(j,t+1) \in Bun(t+1), P(B(j,t+1) | B(i,t)) > 0 \}$$

$$J_a = \{ b : transition \ a \to b \ is \ allowed \} \subseteq \{ b : a \subseteq b \}$$

3.6 Generalised algorithm

The following algorithm is based on a dynamic programming formulation of the stochastic problem, analogous to the one presented for the deterministic problem.

The following definitions will be needed.

Definition 17.
We will call **optesp(a,t,i)** the optimal expected cost from time step **t** to time step **T**, beginning in alternative **a** at time step **t** and knowing that the system is in bundle $B(i,t) \in Bun(t)$.

Definition 18.
A strategy is a set of alternatives
$$str = \{ b(a,t,i) | b(a,t,i) \in J_a, 1 \le a \le nA, 1 \le t \le T-1, 1 \le i \le nb(t) \}$$
The strategy **str** determines the alternative **b(a,t,i)** to be installed from time step **t+1** to time step **T**, if bundle **B(i,t)** would occur and alternative **a** would be installed, at time step **t**.

Algorithm S is designed for the case where scenarios are defined from a little number of possible situations. In this framework scenarios are sequences of situations in time. As we have a reduced number of situations, normally each situation is repeated as part of different scenarios, or as part of the same scenario in different time steps.

For example, a scenario can be a sequence of situations that consist in a high, medium or low level of some relevant variable.

This kind of scenarios are described by the estimated expected operation costs for each alternative, time step and situation, and the sequence of situations that define the scenario. We will then change the definition of a scenario.

Definition 19.
We will call **cost(a,t,s)** the expected operation costs of alternative **a** at time step **t**, given that the system is in situation **s** during all the period **{1,T}**.

Definition 20.
Let **nSC** be the number of considered scenarios. The set of all considered scenarios is represented by a matrix **scen** of dimension **nSC*T** such that **scen(sc,t)** is the situation that corresponds to scenario **sc** at time step **t**.

The following matrices are also calculated, to facilitate the calculations.

Definition 21.
optesp is a matrix of dimension $nA * \sum_{t=1}^{T} nb(t)$. Element **optesp(a,t,i)** will contain the optimal expected cost from time step **t** to time step **T**, beginning in alternative **a** at time step **t** and knowing that we are in bundle **B(i,t)** (also called bundle **i**).

Definition 22.
candi is an **nA*nA** matrix. Element **candi(a,b)** is equal to one if transition between alternative **a** and **b** is allowed, that is (in the framework of algorithm S) if alternative **a** is contained in alternative **b**.

Definition 23.
scen2bun is a **nSC*T** matrix. Element **scen2bun(sc,t)** contains a bundle number **i**. Scenario **sc** belongs to bundle **B(i,t)**.

Definition 24.
sit2bun is a **nSC*T** matrix. Element **sit2bun(i,t)** contains a situation number **s**. For all the scenarios of bundle **i**, situation number **s** is verified at time step **t**. Columns of **sit2bun** are not complete, because **nb(t)<nSC**, except for the last column.

The algorithm that follows is a generalization of the deterministic one, for the stochastic problem.

Algorithm S (stochastic)

1) Generation of scenarios. Includes:
Generation of matrices **scen**, **scen2bun** and **sit2bun**
Generation of vector **nb** (number of bundles for each time step)
Generation of **pr** (probability of each scenario).
Generation of matrix **candi**

2) Initialise
Matrices **str** and **optesp** with zeros.

3) Main process
For t=(T-1:-1:1)
    Calculate matrix $\mathbf{prtr_t}$ for transitions between time steps **t** and **t+1**.
    Optimal expected cost from alternative **a1** and bundle **i1** in time step **t**
    for a1=(nA:-1:1)
        for i1=1:nb(t)
            Load in **J** alternative indices **a2** attainable from **a1.**
            Load in **K** bundle indices **i2** attainable from **i1.**
            For each possible transition to an alternative **a2**, calculate the expected cost for **[t,T]**, given we come from alternative **a1** and bundle **i1** in **t.**
            Calculate the minimal expected cost and its corresponding alternative index **im**. Load **optesp** and **str** from those values.
        end for (i1)
    end for (a1)
end for (t)

4. Examples and applications

Applications present the type of problems that the method described in this paper can handle.

The case of 4.1 can be solved with standard stochastic dynamic programming. In this case there is a standard way to generate many scenarios, and this possibility is used to get an idea about processing time.

The case of 4.2 is closer to real life applications.

Here accuracy is not the main issue. This is due to the inexact estimation of some variables that are known to be relevant, specially the ones related to global economic consequences and to subjective probability of events. In my experience in problems related to the power system, the most important global economic variables are the non-served energy unit price and the alternative investment yield.

Non-served energy unit prices are a set of values (USD/MWh) which are assigned to each unit of non-served energy, in different levels. The unit cost for the first level is much greater than the variable cost of plants in the system, and it increases for deeper levels. These values represent the cost for the economy for not having demand completely fulfilled. They are roughly estimated from studies of the whole economic system. In addition, these studies are expensive, so that results are frequently calculated many years before they are used.

Similar considerations can be made about the alternative investment yield in the economy.

Subjective probabilities of events are other variables of difficult estimation. Information to make it comes from statistical studies, history and information that is not open to the public. In such a situation it is very difficult to even evaluate the degree of accuracy of the resulting estimations.

The use of models permits to take into account all the relevant variables of a system in analyzing a decision. It also permits to react quickly when conditions vary in a significant form.

Finally, the construction of scenarios is made from two or three situations (low, medium and high for example), that is all that can be distinguished at the level of accuracy that we have.

4.1 Markovian scenario

This case have markovian scenarios, drawn from two situations **s1** and **s2**. It has a fixed number of states and a unique square transition matrix. We suppose that the system is in situation **s1** in the first time step. For each time step, probabilities of transition to situation **s1** or **s2** are both 0.5. This problem can be solved using classical dynamic programming, and is used to test calculation time and memory size required.

With a tree of around 1000 scenarios, the problem is solved in less than 5 minutes in a PC Pentium, using a prototype MATLAB code. The aim of this example is to check the code, and to ensure that calculation time is manageable in real life problems.

4.2 An example of generation expansion

A deterministic (second stage) problem results for the analysis of the generation expansion of a power system. Results can be useful for an independent generator (who is thinking of installing power to sell in the spot market), to know the type of plant that is needed in the system in some time step.

An operation model of the system is supposed to be available, as well as the input data that it needs.

The following projects have been considered:

- Four independent and identical gas turbines, identified as TG1 to TG4.
- Two additional identical gas turbines TG5 and TG6, that could be combined.
- The incremental project to get a combined cycle from TG5 and TG6.
- Two projects consisting in the change of fuel of existing units E1 and E2.
- A project of improvement of variable cost and increase of power for E2.

In terms of the deterministic algorithm, each gas turbine is a project, but only 5 alternatives are originated from them (it is known that the first project must be at least of two gas turbines).

The incremental project to get a combined cycle from gas turbines requires the previous turbines to be included in its alternative.

Improvement of E2 was considered an independent project (it could be done with a change in fuel or not).

From the existing projects and the above conditions, 30 alternatives have been considered:

1. No project at all
2. Only change of fuel of E1 and E2
3. Installation of TG5 and TG6 alone (2 TG)
4. 2 TG plus improvement of E2
5. 2 TG and change of fuel of E1
6. 2 TG, change of fuel of E1 and improvement of E2
7. Combined cycle (CC) with TG5 and TG6
8. 1 CC and improvement of E2
9. 1 CC, change of fuel of E1 and E2
10. 1 CC, change of fuel of E1 and E2 and improvement of E2
11. 3 TG alone
12. 3 TG and improvement of E2
13. 3 TG, change of fuel of E1 and E2
14. 3 TG, change of fuel of E1 and E2 and improvement of E2
15. 1 CC and a gas turbine (1 TG)
16. 1 CC, 1 TG and change of fuel of E1 and E2
17. 4 TG
18. 4 TG and improvement of E2
19. 4 TG, change of fuel of E1 and E2
20. 4 TG, change of fuel of E1 and E2 and improvement of E2
21. 1 CC, 2 TG
22. 1 CC, 2 TG and change of fuel of E1 and E2
23. 5 TG
24. 5 TG and change of fuel of E1 and E2
25. 1 CC, 3 TG
26. 1 CC, 3 TG and change of fuel of E1 and E2
27. 6 TG
28. 6 TG and change of fuel of E1 and E2
29. 1 CC, 4 TG
30. 1 CC, 4 TG and change of fuel of E1 and E2

Alternatives with a very high installed power related to the size of the system in time step **T** were eliminated.

This algorithm can run in a PC Pentium, and is written in a student's version of MATLAB. Calculation time is negligible.

The solution for this problem was the following **optim** vector:

$$optim = \begin{bmatrix} 3 & 7 & 7 & 7 & 7 & 15 & 15 & 15 & 21 & 25 & 25 \end{bmatrix}$$

That is, to install TG5 and TG6 the first year and to combine the cycle one year after. Additional gas turbines are needed in years 6, 9 and 10.

Valuable qualitative information can be obtained from the solution of this problem. For example, with this set of data the projects on existing plants are not interesting.

4.3 Generation expansion with uncertainty

Let us call **A** the modeled purchaser, and **B** the other player in the market. **A** and **B** are interconnected, **B** is a seller that has cheap energy. The aim of **A** is to decide how much power he would have to install, having as an alternative to buy to **B**.

The total analyzed period is of 11 years **(T=11).**

This example generates scenarios from three basic situations of trade of electric energy.

Situation 1 (separate systems situation)

There is no trade of electricity between **A** and **B**.

Situation 2 (limited integration situation)

The amount of trade between **A** and **B** is bounded, because **B** decides to limit his offer.

Situation 3 (integrated systems situation)

The amount of trade between **A** and **B** is not limited in quantities. The price of the energy of **B** depends on the alternative that **A** has installed.

Alternatives for **A** consist in installation of generation plants, which require a long period of decision, financing and construction. **B** can choose his prices in a shorter term so that when **B** takes his decisions on prices, he knows what **A** has played. The situation corresponds to a dynamic game with **A** as the first player. This game theory problem can be modelled without changes in the algorithm. When we calculate the row corresponding to an alternative **a**, we know the alternative and then an estimation of the prices **B** is going to choose can be made. Using such prices, we are able to calculate the conditional expected operation cost for each year. Price considerations could be also made in situation 2.

Given a fixed investment level, expected variable costs decrease from situation 1 to situation 3.

The aim of **A** is to calculate his best investment strategy.

Investment alternatives are ordered in fixed cost increasing order. A greater fixed cost corresponds to an increase in installed power or to the installation of lower variable cost generation plants. We suppose that the monotony result holds.

Different expansions result if each situation is considered to hold for all the study period, and there is uncertainty about the sequence of situations that will really occur.

Uncertainty on trade for a 10-years future period can be modelled in many different ways. Even if we construct scenarios from situations outlined above, some sequences of situations can be considered impossible. In other cases, for the sake of simplicity the analyst may prefer to represent a group of similar sequences aggregated in only one scenario with a greater probability.

The analyst knowledge about the system and the information he has about the present situation can produce different probability distributions. So, probability distribution of scenarios represents the (funded) opinion of the analyst about possible futures (including zero probability scenarios, which can be omitted). We think that kind of model is useful to better elaborate opinions, to incorporate verified knowledge about reality and to conserve coherence in decision making.

In this case, we have studied a simple structure consisting of a main scenario and “crisis” scenarios. At each time step, there is a low conditional probability of transition to the crisis scenario given we are in the main one (for the numerical results we have used a transition probability of 0.01). If crisis does not arise, the system goes on in the main scenario. If a transition to a crisis scenario occurs, no new transition occurs until the end of the studied period.

The sequence of situations that defines the **main scenario** begins with **situation 2** in the first time steps, continuing with **situation 3**. This means that the main scenario consist of an improvement in integration between **A** and **B**, that is perceived as having a high probability. If this main scenario were considered as deterministic, a low level of investment would be necessary.

Each crisis scenario coincides with the main one until a transition time step. Then, **situation 1** holds for some years (in the numerical example we have taken 5 years), and then it takes the same sequence of situations as the main scenario. If one of such scenarios were considered as deterministic, a high level of investment would be justified.

Alternatives are labelled from 1 to 11. Labels increase by investment level. Alternative 1 has no additional investment. Investment level is related to installed capacity and variable cost of that capacity.

The following table shows the optimal strategies, obtained using the algorithm presented in this paper. Each column represents a year, and each row is a scenario. The table contents are labels of the optimal alternative for the corresponding scenario and year.

Year	1	2	3	4	5	6	7	8	9	10	11
Main	3	3	3	3	3	3	3	3	5	5	5
crisis 2	3	3	9	9	9	9	9	9	9	9	9
crisis 3	3	3	3	9	9	9	9	9	9	9	9
crisis 4	3	3	3	3	11	11	11	11	11	11	11
crisis 5	3	3	3	3	3	11	11	11	11	11	11
crisis 6	3	3	3	3	3	3	11	11	11	11	11
crisis 7	3	3	3	3	3	3	3	11	11	11	11
crisis 8	3	3	3	3	3	3	3	3	11	11	11
crisis 9	3	3	3	3	3	3	3	3	5	11	11
crisis 10	3	3	3	3	3	3	3	3	5	5	11

As conditional transition probabilities are very low in this example, there is an important probability of remaining in the main scenario for the whole period, and the alternatives for the main scenario can be considered as a plan. The main scenario has no crisis, and its investment level is the lowest. Even though, it is higher than the investment level you would have considering the main scenario as deterministic.

If a crisis scenario have its transition to crisis in a year, then important increases in the level of investment take place the year after. This is intended to support the crisis period. After the crisis period, equipment stay in the system due to irreversibility.

Crisis scenarios 2 and 3 have their crisis early in the study period, so that demand is lower in the crisis period, and Alternative 9 is enough. Crisis scenarios 5 to 8 have a crisis later (with a greater demand), so that they need Alternative 11 investment level.

If the beginning of the crisis is in time step 9 or 10 (as in crisis scenarios 9 and 10) then Alternative 5 is installed as in the main scenario, and investment is completed to the Alternative 11 level after the crisis beginning.

Bibliography

[1] Birge, J.R., Louveaux, F. Introduction to Stochastic Programming. Springer-Verlag, 1997.

[2] Basar, T. Olsder, G. J. Dynamic Noncooperative Game Theory Mathematics in science and engineering – Volume 160. Academic Press INC. (London). 1982.

[3] Nemhauser, G. L. Rinooy Kan, A. H. G., Todd, M. J. Handbooks in Operations Research and Management Science. Volume 1, Optimization. North-Holland 1989.

[4] Rockafellar, R. T. and Wets, R.J.B. Scenarios and policy aggregation in optimization under uncertainty, of Operations Research. Vol. 16, No. 1, (1991) 119-145.

[5] Pereira, M.V.F. and Pinto, L.M.V.G. Multi-stage optimization applied to energy planning, of Mathematical Programming 52 (1991) 359 – 375. North-Holland.

[6] Vázquez, C., Rivier, M. and Pérez-Arriaga, I. J. A market approach to long-term security of supply, of IEEE Transactions on Power Systems, Vol. 17, No.2. May 2002.