

## OPTIMAL ALLOCATION OF RENEWABLE ENERGY PARKS: A TWO-STAGE OPTIMIZATION MODEL\*

CARMEN GERVET AND MOHAMMAD ATEF<sup>1</sup>

**Abstract.** Applied research into Renewable Energies raises complex challenges of a technological, economical or political nature. In this paper, we address the techno–economical optimization problem of selecting locations of wind and solar Parks to be built in Egypt, such that the electricity demand is satisfied at minimal costs. Ultimately, our goal is to build a decision support tool that will provide private and governmental investors into renewable energy systems, valuable insights to make informed short and longer term decisions with respect to park creation and placements. Existing approaches have essentially focused on past data to tackle variations of this problem. In this paper, we introduce a novel approach that considers both past and forecast data, and show the impact for accounting for both sets of data and constraints in a two–stage optimization model. We first show that integer linear programming is best suited to solve the past data model compared to Dynamic Programming and Constrained Local Search. We then introduce our two–stage model that accounts for forecast data as well, adding new constraints to the initial model. Our empirical results show that the two–stage model improves solution quality and overall costs, and can be solved effectively to optimality using Integer Linear Programming.

**Keywords.** Optimization modeling, constraint–based reasoning, park placement problem, renewable energy economics.

**Mathematics Subject Classification.** 68T20, 68W25.

---

Received January 28, 2013. Accepted February 11, 2013.

\* *This research work is partially funded by the STDF grant 1494, Egypt*

<sup>1</sup> German University in Cairo, Tagamoa El-Khamis, New Cairo city 11835, Egypt.  
[carmen.gervet@guc.edu.eg](mailto:carmen.gervet@guc.edu.eg)

## 1. INTRODUCTION

The need for clean energy is recognized worldwide not only to face global warming and  $CO_2$  emissions, but also to reduce grounds for international conflicts. National and international targets are being set [6, 7]. There are many aspects to the development of renewable energy technologies which can be broadly categorized into engineering & technological advancement aspects, versus techno/economical and commercial ones. The engineering components deal with the construction of renewable plants that are reliable, effective and realistic including essentially hydro, solar (photo-voltaic and concentrated solar power), wind and bio-fuel.

The techno-economical study of renewable energy on the other hand, investigates gradual implantation of Renewable Energy (RE) systems for a given country such that the installation and maintenance costs are minimized and the short/long term returns on investment are maximized. Studies in this field investigate country profiles in terms of energy demand, available resources, and anticipated renewable engineering cost reductions [19]. However, more is needed as highlighted in [15], that “there is little economic analysis of renewable energy”. The main objectives of studying the economics of RE is to attract investments (national and international) and set realistic targets and strategies that will remain so in the longer term.

Comprehensive surveys are now available discussing the trends and current improvements in the cost, performance, and reliability of renewable energy systems [6]. Clearly electricity generated from renewable energy sources remains generally more expensive than the one generated from conventional fossil-fuel sources. However, the cost of electricity from RE sources has been falling steadily for the last two decades and various estimates have been derived in terms of “expected cost of electricity production from RE sources” [9]. Today wind energy is the least expensive option but requires lots of maintenance, and is space consuming compared to photo-voltaic solar panels which are, however, currently more expensive. Note that the forecasts in price reduction are promising, as shown in Figure 1 [6]. This indicates that taking into account forecast measurements is a strong element of effective decision making.

Based on existing forecast studies, and each country renewable resources, which REs or portfolio of RE should a given nation invest in? How much should be invested now, in 15 years time? These are questions at the heart of the “economics of RE” for which our decision support tool aims to seek an answer to [15]. We seek the best trade-off cost/return on investment by taking into account physical installation constraints as well as energy requirements and costs. The main beneficiaries will be the governmental as well as private national and international investors.

In this paper we focus essentially on Egypt, even though the methodology employed can be generalized to other countries. The scene in Egypt today can be summarized briefly as follows, regarding the aspects we are concerned with. Energy consumption in Egypt is growing at fast pace and relies extensively on fossil

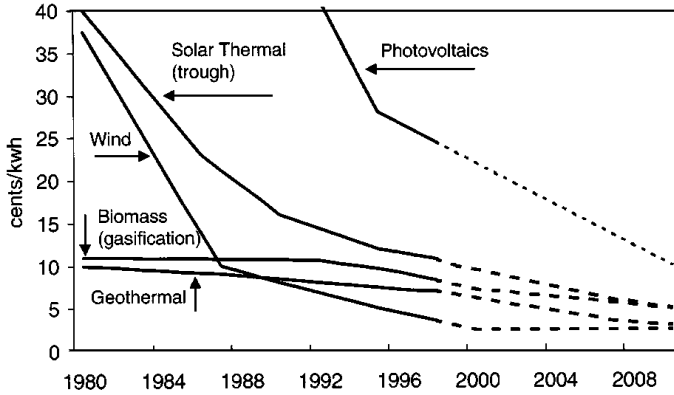


FIGURE 1. RE costs forecast.

fuel as shown in the latest earth-trends survey [29]. Egypt enjoys excellent wind and solar resources and there are tremendous potentials to invest into RE in Egypt, for local consumption and even export. Research and on site projects are being carried out with a growing trend.

A short term plan is to meet 20% of Egypt electricity demand by 2020 using RE sources. While wind farms installation is currently cheaper than solar panels or Concentrated Solar Power (CSP) plants, there are strong arguments in favor of solar energy. Thus it is essential to consider both. Basically solar energy can be installed on small surfaces (little cable and maintenance required), offers more stability especially in Egypt (sun is less dependent on season fluctuations for countries located on the sun belt), plus wasted heat has the advantage of being usable for water desalination [10]. Egypt can become a strong player in wind and solar energy. However, a specific strategy is still to be determined, together with investments. *Our tool aims at contributing towards an efficient strategy into the investment and planning of wind and solar energy parks in Egypt.*

The optimization problem is defined as follows. Given the country of Egypt with its data and constraints: 1) Egypt map of populated areas, and energy demand per area, 2) Wind and Solar atlas, 3) electricity grid map, 4) current and forecast energy cost per RE resource, 5) current and forecast energy demands per month, 6) a set of potential RE park locations; determine the set of energy parks to be invested in today, the set of energy parks to be invested in the future (*e.g.* 5–10 years time), such that atleast 20% of the current and forecast energy demand are covered for each month of the year, and the anticipated financial cost is minimized. The cost is determined in terms of sum of total costs associated with each potential park: cost of connection to the grid, cost of installation, and cost of park maintenance.

In this work, we adopted an iterative development methodology by first focusing on a problem instance with current data only, and extending it to include forecast data and constraints. Thus, both the one-stage and two-stage approaches are

presented. The two-stage approach we introduce relates to two aspects: 1) we address the placement problem by investigating two time-periods simultaneously, 2) we solve the forecast model by exploiting the solution of the problem based on the current data only. One could consider a multi-stage approach that involves more forecast data. However, considering that we are still in a phase in Egypt where no large scale investment has been done, simulating with solutions based on the current data and one forecast data will already be very informative. From a technical point of view, our approach can easily be generalised to a multi-stage model. The only restriction lies in the fact that for each time period considered (*e.g.* 5 years time, 10, 15 etc.), reliable forecast data are made available. Also to remain realistic one should not forecast beyond the limit of investors's span. This explains why in this paper we focus on a two-stage model, that considers current and one 10 years forecast data set.

In this paper, we compare three different models and techniques to tackle the one-stage model: dynamic programming, local search and Integer Linear Programming. A comparison is carried out among all the approaches on different benchmarks and randomly generated instances of the problem. We then apply the most efficient technique to solve the two-stage problem and compare the results of both approaches. This approach is a novel contribution by taking into consideration short term and long term impact on the costs. Using a cost forecast to estimate the cost of the different technologies in the future, our model finds which potential parks should be built now to satisfy the current electricity demand, which should be built after ten years to satisfy a percentage of Egypt's expected electricity demand for the year 2020. To our knowledge this concept has not been considered with respect to renewable energy park placement problems. In other words, our contribution through the models and algorithmic studies, is to allow a decision maker to time his investment and choice of park.

This work was supported by an STDF grant, Egyptian government funding agency. The paper is organized as follows. In Section 2 we survey related works. In Section 3 we introduce optimization techniques that we use to solve the problem. Section 4 describes the data we used to build the prototype and run the simulations. Sections 5 and 6 describe respectively the different models we investigated to solve the first problem. Section 7 defines the forecast model, and Section 8 the implementation. The experimental results are given in Section 9. Finally, a summary of the contribution of this work is given in Section 10 along with suggestions and ideas for future research.

## 2. RELATED WORK

The two-stage approach that takes into account both present and forecast data is novel to our knowledge and has not been addressed with respect to such selection problems involving economical constraints. However, variants of the core problem have been tackled with different constraints and objective functions. We summarize them hereafter. The main model is a form of energy resource allocation

problem using either a single objective (*e.g.* minimizing the total annual cost of building renewable energy parks), or multiple objectives of different sorts such as minimizing pollution emissions of  $CO_2$  and  $NO_x$ , maximize self-production of energy, in addition to the costs factor.

In [23], a survey of different models is presented relative to the problem of energy planning using multi-criteria decision making. While such approaches put a strong emphasis on evaluating the best trade-off between (possibly conflicting) criteria, it does not account for the physical location of parks, nor the future data trends in terms of energy needs and RE sources costs. This could have the downfall that the solution proposed is not technically viable or could be obsolete within few years given the technological advances and cost reductions.

Some approaches address the problem of park placement and selection in specific countries with different objective functions. [21] focuses on the implementation of wind and PV parks to supply electricity in rural areas of Japan. The simulation tool optimizes the cost and seeks to reduce  $CO_2$  emissions. The main drawback is that it only focuses on the present demand and cost values, and ignores the longer term situation. From a different perspective of the problem, [18, 28, 31] focus on the economic dispatch of electricity such that the total fuel cost is minimized, together with the total emissions of  $CO_2$  and  $NO_x$ . As such they do not consider the park placement problem but rather the source of RE to consider to reach the objectives. While such problems have raised a lot of interest due to the study of gas emissions, it does not account of the technical aspects that must be taken into account together with the economical ones.

[2] is the most recent and closest work to ours, where the goal of the decision support tool was to increase renewable energy parks in the US, and in turn reduce the usage of non-renewable source. Similar to our problem, it combines the idea of relating the objective of minimizing costs with the choice of physical location of RE parks, but again no account for future reductions in costs and increased demands. It is important though to note that this work showed empirically that a non-fully utilized park is not cost-effective, mainly due to the fact that a great portion of the cost of establishing a RE park is proportional to the distance of the park to the electricity grid, *ie.* transporting the energy. Thus once the connection is established, one might as well transport as much electricity as possible. We will use this insight in our models.

### 3. TECHNIQUES BACKGROUND

We now briefly recall the foundations of the techniques we will be using in our design implementation, namely dynamic programming, constrained local search and Integer linear programming. Our choice for these techniques was based on our study of the problem components, namely showing the knapsack equivalence, and exploiting the linearity of the constraints, as well as trying to tradeoff efficiency and optimality.

**Dynamic Programming (DP).** DP is a technique occasionally used for solving optimization problems [8]. It amounts to making a sequence of decisions that yield an optimal solution given a cost function. However, not all optimization problems are solvable using DP. The problem must have two main properties namely, optimal substructure and overlapping subproblems. If the problem lacks either of these properties, then it is either unsolvable using DP or inefficient to solve using this technique. The first property ensures that the optimal solution to the problem contains within it an optimal solution to subproblems. By identifying the optimal substructure and thus the subproblems, a recursive formulation of the solution can be defined. The overlapping property ensures that it is worth using DP to solve the problem. When the computation of some states is needed more than once, this is where DP pays off. From an implementation point of view, the time complexity of a DP algorithm can be approximated by the product of the number of subproblems the algorithm goes through overall, and the number of choices necessary to determine which subproblems to use. Examples of problems for which there exists polynomial time DP algorithms are shortest path, sequence alignment, context-free grammar parsing, etc. There are also pseudo-polynomial algorithms to problems such as knapsack, subset sum, cutting-stock [17, 20]. In the next section we show how DP can be used for our problem by reducing it to an instance of the knapsack problem.

**Constrained Local search.** Similar to many other non-deterministic optimization techniques, Local Search (LS) is a method for searching solution spaces of hard combinatorial problems. The result does not guarantee a globally optimal solution. LS basically makes an educated guess to find an initial solution and then makes a fast enough update to reach a better neighboring solution. The main focus of research in this field lies in finding a “good” neighborhood operator, according to the problem structure and objective function. If the problem is constrained and the solution needs to satisfy such constrained we then talk about constrained local search [30]. This is the case in our design of a local search technique for the optimal park selection problem.

**Integer linear programming.** Finally we investigated a third approach that requires the problem to be modeled as a linear problem, ie. all the constraints are linear. The Simplex algorithm is one of the most effective methods used to solve LP. The problem at hand is expressed as a maximization (or minimization) of a linear function, such that all the constraints are linear and the variables are positive reals. If any of the variables take integer values, we are then dealing with an Integer Linear Program (ILP) [25]. This will be the case in our study.

In such cases the Simplex algorithm generally does not yield an integer optimum. Some exceptions can be found though, when the matrix of the model is totally uni-modular yielding a structure whereby the linear programming optimum found by the Simplex method is the integer optimum [16]. However, in the general case further processing is required to reach an integer optimum. The most common techniques, *i.e.* branch and bound and cutting planes, use the Linear Programming (LP) relaxation to estimate the optimal solution of the integer model. The

LP relaxation is the integer problem without the integrality constraints. Briefly, branch and bound branches on a variable which should take an integer value, by dividing the LP problem into two sub-problems, and for each sub-problem adding a constraint on this variable to take a value smaller (respectively greater) than the relaxed value, and solving the LP relaxation for each sub-problem. The order in which the variables are selected is based on various heuristics, the common one being: choose first the variable with non-integral part closest to zero. The best integer solution found provides a bound for the optimal objective value. This technique can be combined with the cutting planes technique, to reduce the search space to be explored without removing any integral solution. Cutting planes are basically extra constraints that “cut off”, part of the feasible region but no integer solution. They can be appended to the model at the beginning or during the optimisation procedure.

## 4. DATA AT HAND

Our core contribution is on the portfolio selection problem for renewable energy parks, such that planning for both short and medium term park installation is seen as one decision problem. One key component to the realistic nature of the model lies in the data made available to us. In this decision support prototype, the data of relevance is as follows, together with its nature in terms of measurement uncertainty.

### 4.1. WIND ATLAS

The wind atlas for Egypt in the form of a hard copy document, was obtained from the NREA (New and Renewable Energy Authority in Egypt), who took measurements over 14 years in partnership with the Danish Ministry of foreign affairs (1991–2005). The outcome provides a comprehensive overview of the wind resource over the entire land area (and sea) of Egypt. The maps produced indicate the mean power density, and wind speeds at a height of 50m over the actual land surface. It gives for a set of specific locations, the prediction of the average yearly energy production of a specific wind turbine or wind farm based on actual measurements of wind speed and direction [14].

### 4.2. SOLAR ATLAS

The solar atlas for Egypt, established in 1998 as part of a USAID project (United States Agency for International Development) and managed by the NREA, is a document including maps of solar energy density for each month. The data is the result of measuring the solar radiation daily in ten locations between 1980 and 1988. The maps were produced using a computerized model that distributes the daily value of global solar radiation into different energy bands [27]. The maps produced as in Figure 3 are monthly average of solar energy density in  $kWh/m^2/day$ .





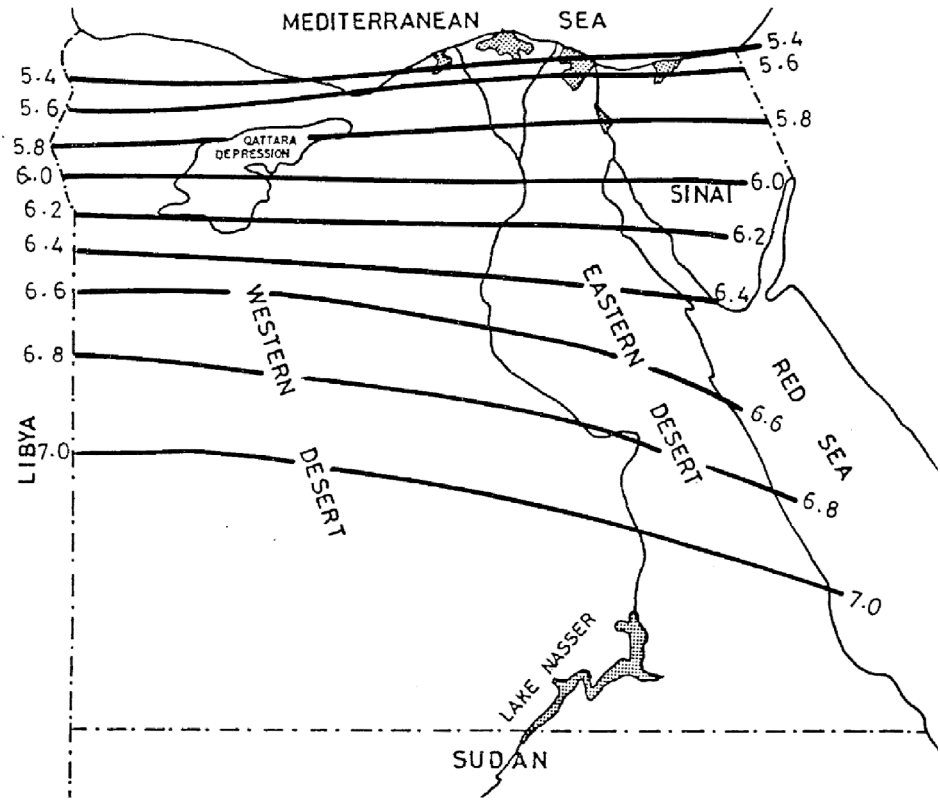


FIGURE 3. Monthly average of the global solar radiation over Egypt in  $kWh/m^2/day$  for March [22].

factor which is directly linked with population growth. Such dynamic parameters can be edited and used to run several simulations and study their impact. The user can currently experiment with the tool in a reverse engineering manner, by seeking whether slightly similar or different outputs are produced from a same set of potential parks but different growth and forecast data.

Also, Egypt does not enjoy yet thorough studies and surveys of fine grained patterns of electricity consumption unlike European countries or North America (e.g. [1]), thus monthly average is the data used. On the other hand the electricity grid in Egypt is a fully connected network and thus when compensation is required to provide more electricity to certain areas like Cairo or Alexandria, this can be achieved and is currently done. It is also important to note that the current plan for renewable energy parks, is to cover at least 20% of the electricity demand by 2020, thus, such parks are not meant yet to be used to cover unforeseen peak demands; rather as a base to off-load the current production based on fossil fuel.

## 5. OPTIMIZATION MODELS

We now explore the approaches attempted to solve the optimal park selection problem. As mentioned earlier, in terms of costs, transportation or connection to the grid is the main bottleneck, thus the problem of selecting a given park, with its maximum capacity, implies that a “yes” is equivalent to “full capacity usage”. This reduces the search space subsequently. Thus independently of the techniques we will use, the model will be Boolean in terms of the decision variables. Another observation that applies to our case study of Egypt, is that the electricity grid is fully connected. This implies that the issue is not about selecting certain park locations to cover certain populated areas, it is more global. As such a traditional set covering model is not required and we introduce our *Grid model*.

### 5.1. THE GRID MODEL

The following input data is provided by the decision maker to our system, but can also be preprocessed and entered automatically.

#### 5.1.1. Input data

$$C_i = \text{Cost of park } i \quad (5.1)$$

$$(x_i, y_i) = \text{Coordinates of park } i \quad (5.2)$$

$$G_{ij} = \text{Watts/m}^2 \text{ produced by park } i \text{ in month } j \quad (5.3)$$

$$D_j = 20\% \text{ Electricity demand in Egypt, month } j, 2010 \quad (5.4)$$

Note that the cost  $C_i$  is the total cost for each potential location and is composed of, installation cost, maintenance cost, and the cost of building the transportation lines to connect and bring the electricity to the grid.

#### 5.1.2. Decision variables

Since a Boolean model is used, the decision variables are associated to each potential park location:

$$\mathbf{B}_i = \begin{cases} 1 & \text{if park } i \text{ is chosen} \\ 0 & \text{otherwise} \end{cases} \quad (5.5)$$

#### 5.1.3. Constraints

The total electricity generated by the RE parks must satisfy 20% of the current demand. Given the demand  $D_j$  for month  $j$ , we have:

$$\sum_{i=1}^n G_{ij} \cdot \mathbf{B}_i \geq D_j \quad \mathbf{B}_i \in \{0, 1\} \quad \forall j \in \{1, \dots, 12\} \quad (5.6)$$

#### 5.1.4. Objective Function

We seek to minimize the current cost over all the potential parks:

$$\text{minimize } \sum_{i=1}^n \mathbf{B}_i \cdot C_i \quad (5.7)$$

### 5.2. MULTI-DIMENSIONAL KNAPSACK EQUIVALENCE

We show that the grid model actually formulates the optimal park selection problem as a Multi-Dimensional Knapsack problem (MDK). This implies that we can benefit from techniques suitable to the MDK to solve our problem. The classical Knapsack Problem (KP) is defined by the tuple  $(n, P, W, Q)$ , where  $n$  elements have each a profit  $P_i$  and a weight  $W_i$ . The objective is to select elements such that the total profit is maximized under the constraint that the total weight should not exceed  $Q$ . We consider the discrete 0-1 KP. The 0-1 KP maps directly our Boolean grid model. More formally the 0-1 KP is specified by:

$$\text{MAXIMIZE } \sum_{i=1}^n P_i \cdot \mathbf{X}_i \quad (5.8)$$

$$\text{subject to } \sum_{i=1}^n W_i \cdot \mathbf{X}_i \leq Q, \quad \mathbf{X}_i \in \{0, 1\} \quad (5.9)$$

where  $\mathbf{X}_i = 1$  if and only if the  $i^{\text{th}}$  element is selected.

The multi-dimensional KP extends it and is specified as a tuple  $(n, m, P, W, Q)$  where  $m$  is the number of dimensions; and we have:

$$\text{MAXIMIZE } \sum_{i=1}^n P_i \cdot \mathbf{X}_i \quad (5.10)$$

$$\text{subject to } \sum_{i=1}^n W_{ij} \cdot \mathbf{X}_i \leq Q_j, \quad \mathbf{X}_i \in \{0, 1\}, \forall j \in \{1, \dots, m\} \quad (5.11)$$

where  $W_{ij}$  is the weight consumed by the  $i^{\text{th}}$  element in the  $j^{\text{th}}$  dimension, and  $Q_j$  is the limit of the  $j^{\text{th}}$  dimension. Note that there is one constraint per dimension. A solution to the MKP is one that satisfies the conjunction of the  $m$  constraints (5.11). An optimal solution is one that maximizes (5.10).

Obviously, the objective of the standard MKP problem is to maximize the profit, whereas in the optimal park selection problem our objective is to minimize the total cost. Also, in the MKP the constraints set an upper bound not to be exceeded for each dimension, whereas in the park selection we have a lower bound to cover the electricity demand for each month. We just need to transform the problem by multiplying constraints and objective by  $(-1)$  to have the equivalence. The optimal park selection problem corresponds to the MKP, whereby  $D_j = -Q_j$  is the demand to be covered at month  $j$ ;  $G_{ij} = -W_{ij}$  is the energy gain of park  $i$  during month  $j$ ; and  $C_i = -P_i$  is the total cost of park  $i$ .

## 6. ALGORITHMS

We present the different algorithms we implemented to solve this problem. The first is a pseudo-polynomial deterministic algorithm that finds the exact optimal solution using DP. The second is a constrained local search algorithm that uses a neighborhood operator to find a good solution. Complexity results are given not so to compare algorithms which are not comparable indeed (complete versus incomplete ones) but rather to show the parameters that matter for a given algorithm. The third and most successful approach uses Integer Linear Programming.

### 6.1. DYNAMIC PROGRAMMING ALGORITHM

The most common approach to solving the KP and MKP is dynamic programming [20]. Thus we first extend it to the the optimal park selection problem. Given  $n$  potential park locations to choose from, and assume for simplicity a constant monthly demand of  $d$  kw/h, we can divide the problem into two independent subproblems. In the first subproblem, the  $n^{\text{th}}$  location is included in the solution, the cost associated with the  $n^{\text{th}}$  is subtracted in the new subproblem and the cost is added. In the second subproblem the  $n^{\text{th}}$  location is not included. The following smaller subproblems solve the same problem with  $n - 1$  potential locations. More formally, the problem to be solved is defined recursively by:

$$(n, C, G, d) \tag{6.1}$$

where  $n$  is the number of locations,  $d$  is the rate of electricity required in kw/h,  $G_n$  is the production gaing of kw/h provided by the  $n^{\text{th}}$  potential location, and  $C_n$  is the total cost associated with it. The objective function to minimize is  $F(n, d)$  which is defined recursively by:

$$F(n, d) = \begin{cases} 0 & \text{if } d \leq 0 \\ \infty & \text{if } n = 0 \text{ and } d > 0 \\ \min \left( \begin{array}{c} F(n - 1, d) \\ C_n + F(n - 1, d - G_n) \end{array} \right) & \text{otherwise} \end{cases} \tag{6.2}$$

Clearly if the demand  $d \leq 0$  then the cost is 0, while for strictly positive demand and no potential locations, there is no solution which is represented by an infinite cost value. Otherwise, the solution is the minimum of two subproblems, one that includes the  $n^{\text{th}}$  location and the other that excludes it.

#### 6.1.1. Complexity

This recursive formulation leads to a DP algorithm with a time complexity expressed in terms of:  $n$ , the number of potential location,  $D_j$  the demand at month  $j$  (in  $\{1, \dots, 12\}$ ), and  $d$ , the largest demand to be covered. As aforementioned with respect to knapsack problems, the time complexity is pseudo-polynomial, which is theoretically the best available for the classical KP. The time complexity is in  $O(n \cdot d^m)$  where  $d = \max_{j=1}^m D_j$ .

In our version of the problem  $m$  is a constant equal to 12, corresponding to the number of months in this case. The same bound of the time complexity serves as a loose upper bound on the space complexity. This is due to the fact that when top-down DP is used, some subproblems are never reached during solving the original problem. The reason for that is the first base case of recurrence (6.2), which prunes the computation once the demand has been satisfied. Computational results are presented in the implementation section.

### 6.1.2. Preprocessing: adding a sorting heuristic

we also considered a preprocessing step to this approach, by sorting the locations before running the DP algorithm. The sorting rule is defined as follows:

$$\text{Location}_a < \text{Location}_b \iff \sum_{j=1}^m G_{aj} < \sum_{i=1}^m G_{bi}$$

This causes the locations with higher gain values for electricity rates to come first in the list of locations. Such an ordering forces many branches of the search tree to be pruned early, hence a better performance. Note that this does not affect the optimality of the algorithm. In other words, the sorting heuristic does not affect the completeness of this approach. However, it does make it more efficient in practice. In Section 9, the sorting heuristic is evaluated, where we compare the running times of two versions of the algorithm, one which applies the sorting heuristic, and one which does not.

## 6.2. CONSTRAINED LOCAL SEARCH ALGORITHM

While the sorting heuristic improves the running time of the DP approach, it is not effective enough to scale up the problem as we illustrate in the experimental results. To improve, we sought a polynomial-time, but sub-optimal algorithm to solve the problem. Now we discuss a local search method used to determine a sub-optimal solution to the problem. The main aspect of local search techniques is to define the neighborhood operator. We use a successful neighborhood operator suggested by Ghosh *et al.* [12]. Their work focuses on solving the subset sum problem and comparing their suggested neighborhood with previous ones for the same problem. The reason we use a subset sum problem technique to solve our problem, is that the MKP model is very similar to subset sum with a slight change in the objective function. In MKP the chosen elements in the solution may not exceed the upper bound, but in the subset sum problem the chosen elements must sum up exactly to the bound.

In our work, we modify and implement the idea to solve the optimal park selection problem as a MKP, which requires several changes to [12] that are highlighted here.

### 6.2.1. Neighborhood operator

In order to define the operator we first define the permutation we use and how a solution is specified. We define  $Y$  the permutation vector over all potential parks. Instead of reasoning directly about the Boolean vector  $\mathbf{X}$ , and trying to improve upon a solution using operations on  $\mathbf{X}$ , we represent a search space of permutations  $Y$  of the park location indices. The neighbor of a permutation  $Y$  is a vector that differs from  $Y$  in only two values (one swap). For example, for  $n = 4$ , and  $Y = [1, 2, 3, 4]$ , we have  $\text{NEIGHBORHOOD}(Y) = \{\underline{[2, 1, 3, 4]}, [3, 2, \underline{1}, 4], [\underline{4}, 2, 3, \underline{1}], [1, 3, \underline{2}, 4], [1, \underline{4}, 3, \underline{2}], [1, 2, \underline{4}, 3]\}$ . However,  $[3, 1, 2, 4]$  and  $[4, 3, 2, 1] \notin \text{NEIGHBORHOOD}(Y)$  since they each contain two swaps.

For a given permutation vector  $Y$ , we denote  $Y_i$  the value of its  $i^{\text{th}}$  argument. For instance for  $Y = [2, 1, 3, 4]$ ,  $Y_2 = 1$ . For sake of notation simplicity, we will perform vector operations and denote by the vector  $G_i$  the energy coverage of park  $i$  (over the 12 months period). In other words it represents the row  $i$  of the matrix  $G_{ij}$ .

---

#### Algorithm 6.1 GREEDY

---

**Require:**  $G, D, Y$

$\mathbf{E} \leftarrow \{\mathbf{E}_i = 1, \forall 1 \leq i \leq n\}$  {solution vector with all parks selected}

$V \leftarrow [V_1, \dots, V_m]$  such that  $V_j = \sum_{i=1}^n G_{ij}$

{vector of aggregated energy covered by all parks per month  $j$ }

**for**  $i = 1 \rightarrow n$  **do**

**if**  $V - G_{Y_i} \geq D$  **then**

$V \leftarrow V - G_{Y_i}$

$\mathbf{E}_{Y_i} \leftarrow 0$  {park  $Y_i$  is removed from the solution vector  $\mathbf{E}$ }

**end if**

**end for**

**return**  $\mathbf{E}$

---

The local search algorithm operates as follows: choose an initial permutation randomly, use the greedy heuristic to compute the solution that matches the permutation with a good cost while satisfying the constraints ( $\text{GREEDY}(G, D, Y)$ ), compute the set of neighboring vectors, determine if a neighbor point can lead to a better solution cost ( $\text{FINDBEST}(Y)$ ). If one exists, move to this permutation with associated solution, if not exit.

The key point lies in the heuristic used to compute a good solution for a given permutation. The greedy algorithm starts by including the entire list  $\mathbf{E}$  of potential locations in the initial solution. Given a permutation  $Y$  of the potential locations, the algorithm traverses the potential locations according to the ordering of  $Y$ . For every potential location, if removing it from  $\mathbf{E}$  causes the total demand to be unsatisfied for at least one of the 12 months (*i.e.* the vector difference  $V - G_{Y_i} \geq D$  is false), the algorithm does not remove it, otherwise it is removed from  $\mathbf{E}$ . The selection of park  $Y_i$  in the current solution vector  $\mathbf{E}$  is disabled,  $\mathbf{E}_{Y_i} = 0$ ; and we

need to cover the demand vector  $D$  with the coverage offered by the remaining parks (updated vector  $V$ ).

---

**Algorithm 6.2** Local Search
 

---

**Require:**  $G, C, D$

$Y \leftarrow [Y_1, Y_2, \dots, Y_n]$  {random permutation}

$X \leftarrow \text{GREEDY}(G, D, Y)$

**loop**

$Y' \leftarrow \text{FINDBEST}(Y)$  {finds the best permutation in the neighborhood of  $Y$  which is better than  $Y$ }

**if**  $Y'$  does not exist **then**

**return**  $X$

**end if**

$X \leftarrow \text{GREEDY}(G, D, Y')$  {binary solution from the greedy algorithm (6.1)}

**end loop**

---

The local search algorithm moves away from an initial permutation  $Y$  by searching for the best neighboring permutation  $Y'$ , that corresponds to the satisfiable neighbor with the smallest cost. The algorithm terminates after failing to find a better permutation to switch to. Typically, a local search algorithm is run for a number of iterations, and the final result is the best solution over all solutions which the algorithm terminated with during all the iterations. At every new iteration in our implementation, the algorithm is not permitted to re-visit permutations that were already encountered during previous iterations<sup>2</sup>.

### 6.2.2. Complexity

The highest cost of Algorithm 6.2 is **FINDBEST**, which finds the best next solution, better than the current one from within the neighborhood. The time complexity of this **FINDBEST** is polynomial in the number of locations.

$$O(n^3 \cdot m) \tag{6.3}$$

where  $n$  is the number of potential locations, and  $m$  is the number of seasons (12). Therefore,  $m$  is a constant and the complexity can be reduced to  $O(n^3)$ .

The time complexity of the full algorithm depends on how many iterations the loop in Algorithm (6.2) does. This is controlled by how far is the initial random permutation from a local minimum, since the algorithm terminates when no better solution is found in the neighborhood.

## 6.3. INTEGER LINEAR APPROACH

Finally we tried a third approach which proved the most effective in terms of both, solution quality and efficiency. The Boolean variables in Equation (5.5),

---

<sup>2</sup>This is called local search with memory, which resembles the operation of tabu search.

TABLE 1. All possible scenarios of building  $a$  and  $b$ .

	Build park $a$	Build park $b$	Demand 2010	Demand 2020
1	no	no	no	no
2	no	yes/2010	satisfied	no
3	no	yes/2020	no	no
4	yes/2010	no	satisfied	no
5	yes/2020	no	no	no
6	yes/2020	yes/2020	no	satisfied
7	yes/2010	yes/2010	satisfied	satisfied
8	yes/2010	yes/2020	satisfied	satisfied
9	yes/2020	yes/2010	satisfied	satisfied

$X_i$  lie in the domain  $\{0, 1\}$ . The objective function is a linear sum of the costs of all potential locations chosen to be in the solution. Also, the constraints in Equation (5.6) are linear constraints ensuring that the demand for each month is satisfied by the chosen potential locations. This grid model is now solved using the Simplex algorithm to find an optimal solution which would in general violate the integrality constraint on the variables. While the Simplex algorithm might not produce an integral solution, the solution to the LP relaxation is used to direct the search towards an integer one using a complementary technique called *branch and bound*, briefly described in Section 2. In our case the branching will create two sub-problems when handling each 0–1 decision variable  $X_i$ , by constraining the variable to take a value  $\leq 0$  or  $\geq 1$  thus forcing the Simplex method to assign 0 or 1 to such decision variable when seeking an optimum solution to each sub-problem. The CPLEX tool contains a branch and bound technique to tackle integer or mixed integer linear programs.

In our case, solving the grid model directly with a standard ILP solver proved to be the best technique in terms of CPU and optimality. Thus we use it for the two-stage optimization model.

## 7. TWO-STAGE OPTIMIZATION MODEL

In this section, we extend the grid model (Sect. 5), to include forecast data, namely the costs of potential parks and expected electricity demand for 2020. The impact of considering the two-stage model is illustrated in the example below. Let  $a$  and  $b$  be 2 potential renewable parks. Assume a monthly demand with an annual growth rate, and assume that each park can satisfy alone the present demand but not the forecast one. We have the following combinations of scenarios shown in Table 1, of which 3 satisfy all the constraints and the optimal one shall be the one with lowest cost that relies on forecast cost reductions (either scenario 8 or 9 depending on the cost values).



The uncertainty lies in the technologies costs, as well as the forecast electricity demand for 2020. It can be estimated with existing forecasting curves of market and demographic studies. It is expected as shown in Figure 1, that the cost for renewable energy technologies will decrease in the coming years [6]. For each type of uncertainty, the current approach considers an annual growth rate for the electricity demand (that can be tuned by the end-user), and a total future cost per potential park according to the technology used and its transportation cost. Thus the initial grid model is extended with the following underlined terms and equations. The objective now is to find the minimum total cost for satisfying the demand of today and tomorrow, by taking into account forecast values.

The input for the final model is:

$$C_i = \text{Cost of park } i \quad (7.1)$$

$$\underline{FC}_i = \underline{\text{Future cost of the } i^{\text{th}} \text{ location}} \quad (7.2)$$

$$(x_i, y_i) = \text{Coordinates of park } i \quad (7.3)$$

$$G_{ij} = \text{Watts/m}^2 \text{ produced by park } i \text{ in month } j \quad (7.4)$$

$$D_j = \text{Electricity demand in 2010, month } j \quad (7.5)$$

$$\underline{r} = \underline{\text{Annual growth rate of electricity demand}} \quad (7.6)$$

$$\underline{FD}_j = \underline{D_j \times r \text{ future demand, month } j, 2020} \quad (7.7)$$

**Model Variables.** We extend the set of variables from the short-term model, with new ones as shown below.

$$\mathbf{X}_i = \begin{cases} 1 & \text{The } i^{\text{th}} \text{ location is to be built in the present} \\ 0 & \text{otherwise} \end{cases} \quad (7.8)$$

$$\underline{\mathbf{F}}\mathbf{X}_i = \begin{cases} 1 & \text{The } i^{\text{th}} \text{ location is to be built in the future} \\ 0 & \text{otherwise} \end{cases} \quad (7.9)$$

**Constraints** The integer linear program has the following linear constraints:

$$\sum_{i=1}^n G_{ij} \cdot \mathbf{X}_i \geq D_j, \quad \forall j \in \{1, \dots, m\} \quad (7.10)$$

$$\sum_{i=1}^n G_{ij} \cdot (\mathbf{X}_i + \underline{\mathbf{F}}\mathbf{X}_i) \geq \underline{FD}_j, \quad \forall j \in \{1, \dots, m\} \quad (7.11)$$

$$\underline{0 \leq \mathbf{X}_i + \underline{\mathbf{F}}\mathbf{X}_i \leq 1, \quad \forall i \in \{1, \dots, n\}} \quad (7.12)$$

$$\mathbf{X}_i \in \{0, 1\}, \quad \underline{\mathbf{F}}\mathbf{X}_i \in \{0, 1\}, \quad \forall i \in \{1, \dots, n\} \quad (7.13)$$

Constraints 7.10 ensure that the locations chosen to be built in the present satisfy the current demand, while constraints 7.11 ensure that all chosen locations built in the present and the future satisfy the future demand as well. Constraints 7.12 prevent each potential location from being built twice.

**Objective Function.** The objective function defined in the approach is obviously a linear one. Since we seek to minimize the sum of all costs, given by the expression:

$$Cost = \sum_{i=1}^n \mathbf{X}_i \cdot C_i + \sum_{i=1}^n \mathbf{F}\mathbf{X}_i \cdot FC_i \quad (7.14)$$

The cost expression is the sum of two scalar products, one for the locations chosen to be built today, and the other for the locations chosen to be built in the future. This model is also solved using the Simplex algorithm, with built-in branch and bound to search for an integer optimum.

## 8. IMPLEMENTATION

We discuss the implementation of the different algorithms used and our prototype. Then we analyze the results obtained. The entire process of implementing all models was done on an Ubuntu 10.04 (Linux-based) Operating System. We first highlight the sources of the data used during testing and evaluation. The DP algorithm (6.2) as well as the constrained local search one (6.2), were implemented in C++. The DP algorithm uses top-down DP. Caching the intermediate solutions for the subproblems is done using the `map` data-structure in the C++ STL. The memoization is achieved by keeping track of an array of `maps`. Each `map` stores key-value pairs, where the key is STL's `vector` container, representing the total gain vector  $\mathbf{x}$  and the value is the solution to the corresponding problem. The constrained local search algorithm was stopped when the optimum was found (in our case the LP solution). The ECL<sup>i</sup>PS<sup>e</sup> [5] platform was chosen to implement the ILP model thanks to its high level of abstraction and hybridization libraries with the `cplex` solver.

### 8.1. DATA PRE-PROCESSING

We purchased from a governmental body the wind and solar solar atlas of Egypt [14, 22]. In the form of printed documents and maps, they contain detailed information about solar radiation, wind speed, wind directions for each month of the year for a given location (resulting from analysis over the past 8 years). There are 12 maps in the solar atlas, each specifying an amount of watts/m<sup>2</sup> for each month for a given latitude. In the solar atlas, the annual average rate of electricity is given in a single map. A sample of the solar map for one month was illustrated earlier in Figure 3. The information in these maps was extracted using Python's image processing library SciPy [26]. We scanned the atlases, then processed the maps' images by labeling the different regions over each map with their corresponding solar radiation or wind power [11]. The result of the extraction process was the generation of a function that takes as input  $(x, y, t)$  and returns the kWh value, representing the gained rate of electricity at location  $(x, y)$  during month  $t$ .

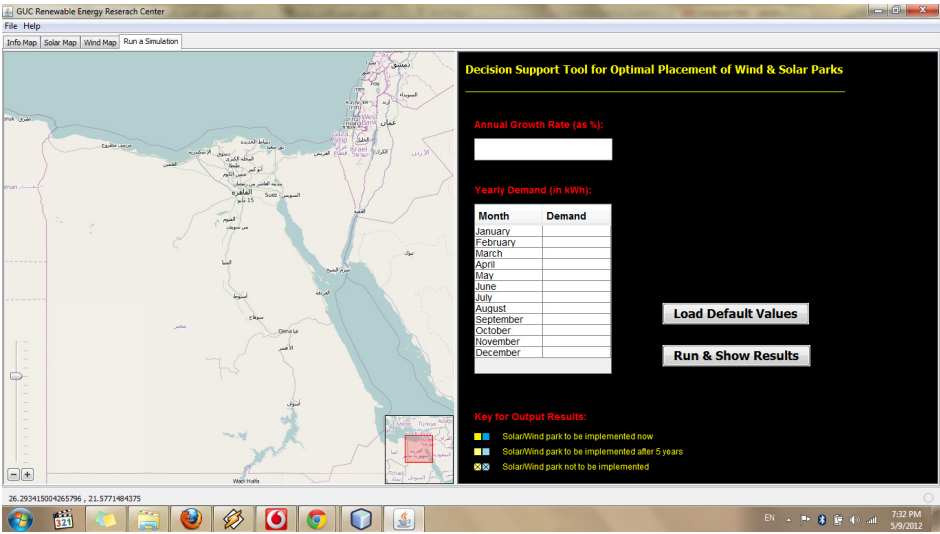


FIGURE 4. Home page of the GUI to enter relevant data.

### 8.1.1. Graphical user interface

The GUI module was implemented using Java. In order to enable the user to freely select potential locations from the map of Egypt, the OpenStreetMap [13] library was used. The user inputs data regarding the cost and maximum area for each location selected on the map. The database contains default data for the current demand of Egypt as well as the forecasted cost data shown in Figure (1). This information is stored and available for the solver to read during execution.

### 8.1.2. Instance data sets

For a given run of the tool, a dataset is composed of  $n$  points chosen by the user, on a digitalized map of Egypt. The energy gain of these locations is derived from the gain matrix  $G$  for the dataset. However, the costs for these locations is currently set by the user through the GUI. It reflects on the type of the park (wind, solar PV, solar CSP), its size and distance to the grid.

Regarding the demand vector, we generated random instances to better evaluate the different techniques, as follows:

$$D_j = F \cdot \sum_{i=1}^n G_{ij} \quad (8.1)$$

Basically, each value  $D_j$  in the demand vector is computed by multiplying a factor  $F < 1$  by the sum of all  $G_{ij} \forall 1 \leq i \leq n$ . This factor  $F$  was devised, such that it determines the level of difficulty for the demand to be satisfied by the potential parks energy gain. This difficulty level relates to the tightness of the

model to satisfy the demand given the number of potential parks and respective energy gain. The closer  $F$  is to 0.5 the largest the number of combinations of potential parks need to be considered since it implies on average that 50% of the parks will be needed to cover the demand. For example, if  $F = 0.3$  and  $n = 100$ , then about 30 parks will be needed to satisfy the given demand.

## 9. EVALUATION

We first present the results of evaluating the scalability of the different algorithms we evaluated: DP, DP with preprocessing using sorting (DPS), constrained Local search (LS), and ILP. We show the results on the average running time of 20 instances of the problem chosen randomly. This average is taken over all instances of the same problem size,  $n$ . Note that we stop the constrained local search when it reaches the ILP optimum, which was the case in these runs.

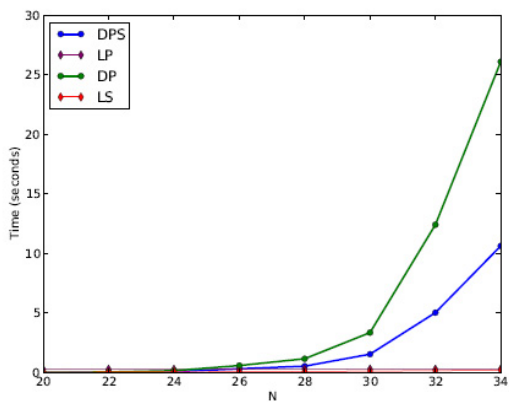
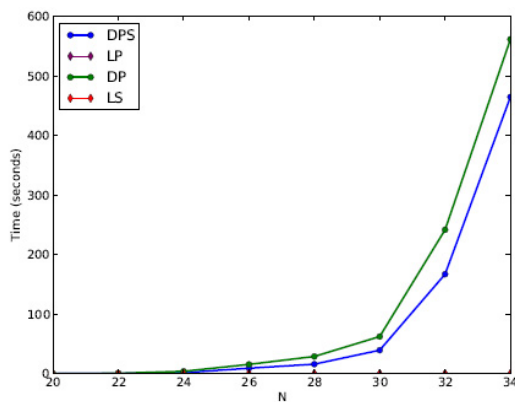
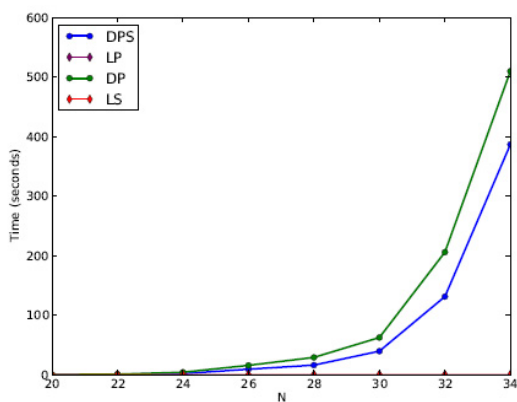
In Figure 5 we show the CPU running times of all approaches for a group of datasets where  $20 \leq n \leq 34$  and  $F \in \{0.25, 0.5, 0.75\}$ . DP and DPS have very high running times according to the shown results. As we can see from the Figure 5a, the running times of DP and DPS do not exceed 50 seconds for  $n \leq 40$ . However when  $F \geq 0.5$ , the worst running time becomes quite high. Also, the DP and DPS approaches tend to require memory more than 2GB for  $n > 30$ . The approaches that proved efficient for this problem size are LS and ILP. This is expected, since DP and DPS have a pseudo-polynomial running time. When the required potential locations in the solution  $k$  approaches  $F \times n$ , the number of solutions considered grows rapidly, which affects the running time of DP and DPS approaches.

Now we show the performance of the LS and ILP algorithms for larger values for  $n$ ,  $40 \leq n \leq 220$  (Fig. 6). The numbers are actually the same for the different values of  $F$  we considered, thus the main issue here is on the number of potential parks to consider rather than the tightness of the constraints. The results show a polynomial increase in the running time with respect to  $n$ . This reflects the  $O(n^3)$  time complexity of the LS approach used. On the other hand, the ILP approach proves to be extremely fast and robust in solving all instances.

### 9.1. FORECAST ILP MODEL

The solution was broken into two stages. First the simple gridmodel described in Section 5 is solved whereby the selection of parks at present should cover 20% of the forecast demand directly. The two-stage model builds on this first solution by constraining the total cost to be smaller than the one produced by the grid model. It becomes an upper bound to the extended cost function since we can ensure that future costs are cheaper than covering the forecast demand from the present investment.

We show that adding this constraint and solving the two-stage problem with the constrained new objective function (7.14), even though it requires a second run of the Cplex solver, it greatly speeds up the runtime of the entire algorithm.

(A)  $F = 0.25$ (B)  $F = 0.50$ (C)  $F = 0.75$ FIGURE 5. Runtime of DP, DPS, ILP, and LS approaches for  $20 \leq n \leq 34$ .

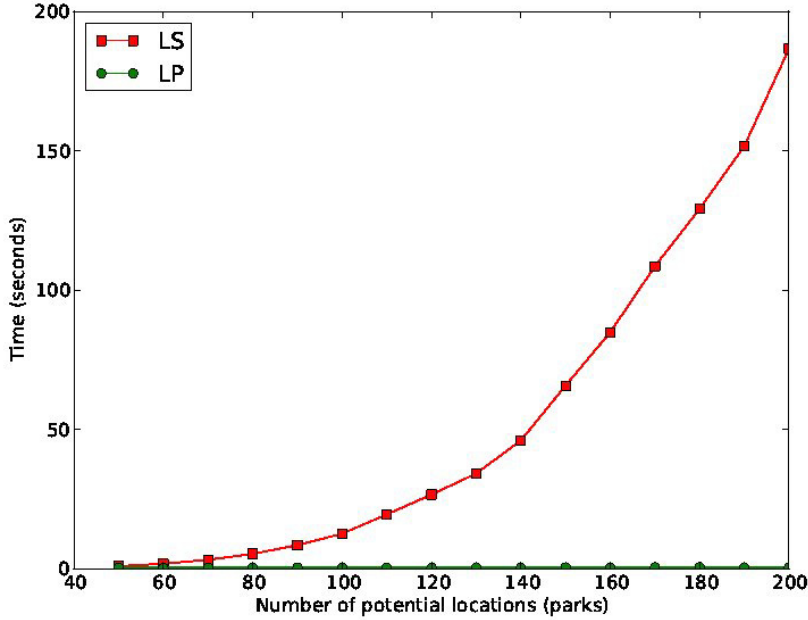


FIGURE 6. Runtime of ILP and LS approaches for  $40 \leq n \leq 220$

In other words, solving the forecast model without the upper bound constraint on the cost is less efficient than solving grid model, acquiring the upper bound, then solving the forecast model, as shown in Figure 7. The testing was done for larger number of potential locations,  $50 \leq n \leq 250$  given its efficiency.

The performances show that the LP approach offers efficiency as well as robustness to compute the optimal solution. Even though we are dealing with a simulation and a decision support tool, where real time and computational time is not the first factor to consider when choosing an algorithm, this study shows the suitability of the LP approach. Indeed, the LP approach guarantees optimality (an important element of confidence in the tool for decision makers), is efficient, scaleable, and robust to changes in the model, including the forecast ILP.

## 9.2. SOLUTIONS PRODUCED

We now illustrate the impact of reasoning with forecast demand and cost values in terms of the solution quality and difference in final parks choice. A set of 10 potential solar and wind parks are initially placed on the GUI. These two sets are chosen by randomly selecting coordinates on the map of Egypt. The output of the algorithm, is the optimal set of parks to be built in the present or in 10 years time in order to satisfy both demands. In Figure 8, the optimal set is shown when only present data is considered. In Figure 9, it is shown how the optimal solution differs when we account for forecast costs and demands. We can see that solar park 8 and

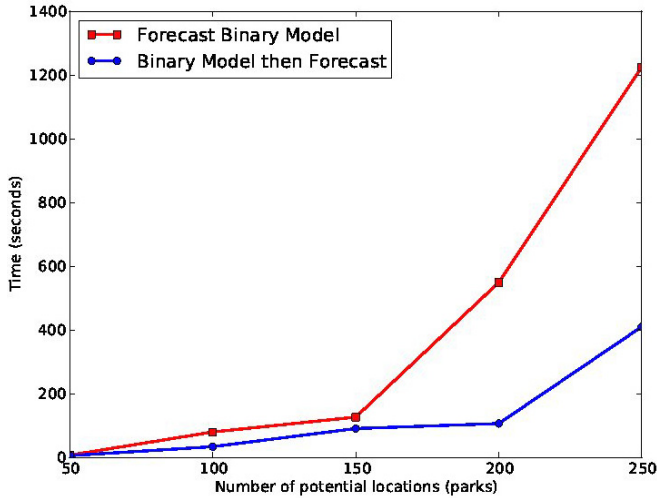


FIGURE 7. Runtime of the ILP long term model for  $50 \leq n \leq 250$ .

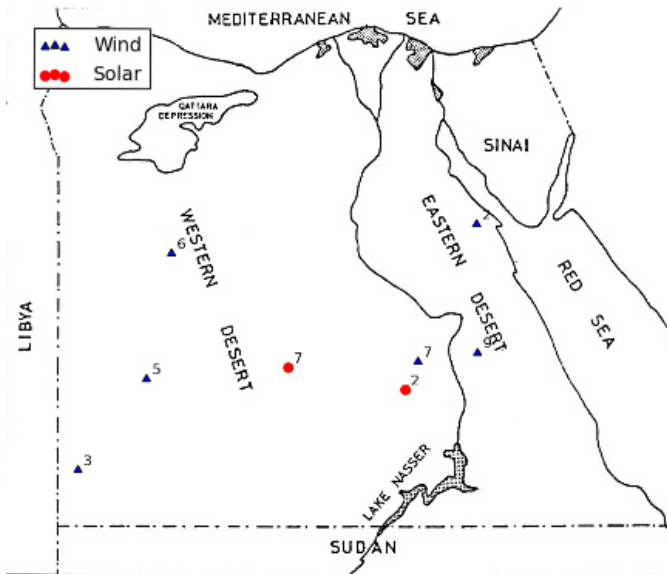


FIGURE 8. Chosen locations using grid model.

wind parks 1 and 7 are best to be built in the future and wind park 5,7 and 9 for instance should not be considered at all when future data is taken into account.

We can notice that the selection of parks and the planning regarding when to invest, differs. One can see, in particular, the shift between the first solution where

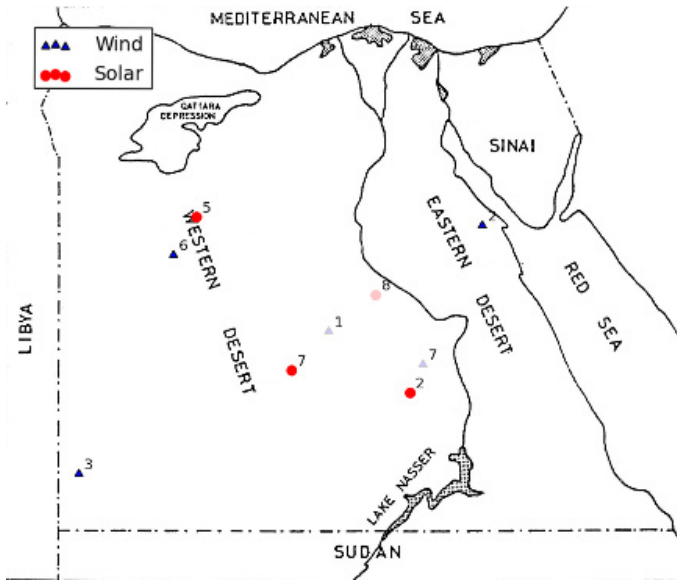


FIGURE 9. Chosen locations with two-stage model.

more investment is put into wind farms, and the second one where solar parks become predominant with identical input of potential parks.

The solutions differ also in terms of the costs of investment when the number of chosen parks grows. With less than ten parks the costs are similar, however as we simulated with a much larger number of parks both CSP solar and wind, the cost difference increased, while maintaining the constraints of covering both the present and forecast demands. In other words, we run what we call the binary model that relies solely on the investment in parks at present ( $B_i$  variables) to cover both present and future demand, and compared its total cost with the forecast model. The difference in cost was anticipated given the reduction of parks' costs over time, however its order of magnitude shows the importance of timing the investment (cost expressed in 1000 of Egyptian Pounds). This is illustrated in Figure 10.

## 10. CONCLUSION

In this paper we have extended the state-of-the-art models for optimal park selection, to account for both current and forecast data in the decision making process. We showed how this impacts greatly the solution quality, and also how it can be achieved efficiently using a two-stage approach as an ILP model.

Clearly, when dealing with forecast estimates, even from domain experts, uncertainty is present and cannot be overlooked. While the scope of this paper is to study the impact and added value of our two-stage model, part of future work on this project is to extend the linear model we obtained by considering the parameter



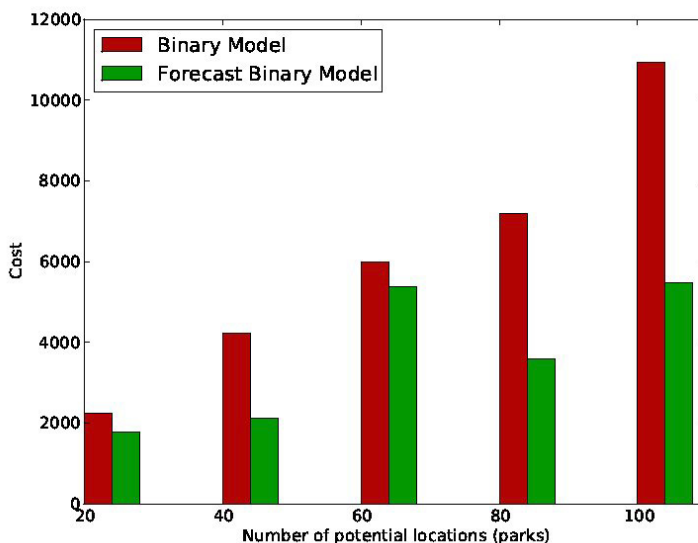


FIGURE 10. Solution costs comparison.

data as interval coefficients in the model, thus introducing a robust enclosing of the uncertainty measures, to seek a reliable constraint model. Interval linear models have shown their success and suitability to reason with incomplete and erroneous data (*e.g.* [3]). Furthermore, our previous work in this field, including our more recent work on CDF-intervals [24], and in robust optimization in general have shown that interval linear models can be very effectively transformed into tractable linear models and solved using LP techniques to derive solution sets [4,32]. The ECL<sup>i</sup>PS<sup>e</sup> platform now supports such coefficients format in its modelling environment [5]. Thus, future work includes the migration of the linear model to a reliable interval models to account for erroneous data from the forecast data specified by the user. Also, given the strong interests from domain experts in Egypt towards the potential of such a tool, we definitely intend to extend the actual problem specification to finer grained level of constraints. For instance by modelling the reduction of a park production over time, to account for a more accurate production rate per month of a given time period.

## REFERENCES

- [1] *Model documentation electricity capacity planning submodule of the electricity market module*. Technical report, Nuclear and Electricity Analysis Branch, Energy Supply and Conversion Division, Office of Integrated Analysis and Forecasting, Energy Information Administration (1994).
- [2] A.N. Arnette, *A Spatial Decision Support System for the Development of Multi-Source Renewable Energy Systems*. PhD thesis, Virginia Polytechnic Institute and State University (2010).
- [3] Y. Ben-Haim and I.Elishakoff, *Convex Models of Uncertainty in Applied Mechanics*. Elsevier (1990).

- [4] A. Ben-Tal and A. Nemirovski, Robust convex optimization. *Math. Operat. Res.* **23** (1998) 1–38.
- [5] P. Brisset, H. El Sakkout, T. Fruewirth, C. Gervet, W. Harwey and M. Meier, Le Provost T. Novello, S., J. Schimpf, K. Shen and M. Wallace, *Eclipse constraint library manual*. Technical report (2012).
- [6] S.R. Bull, Renewable energy today and tomorrow. *Proceed. IEEE* **89** (2001) 1216–1226.
- [7] S.R. Bull, The integration of renewables, in *the First international conference on the integration of renewable energy sources and distributed energy resources, Brussels, 1-3 December 2004*.
- [8] S. Dreyfus, Richard bellman on the birth of dynamic programming. *Operat. Res.* (2002) 48–51.
- [9] *Energy efficiency ad Renewable energy*. [www.eere.energy.gov/wind](http://www.eere.energy.gov/wind).
- [10] M.N. El-Kordy, M.A. Badr, K.A. Abed and S.M.A. Ibrahim, Economical evaluation of electricity generation considering externalities. *Renewable Energy* **25** (2002) 317–328.
- [11] Saher El-Neklawy. *Extracting solar radiation data from egyptian solar atlas*. Technical report, GUC (2011).
- [12] D. Ghosh, N. Chakravarti, SS Fatima, M. Wooldridge, and NR Jennings. A competitive local search heuristic for the subset sum problem. *Comput. Operat.s Res.* **26** (1999) 271–280,.
- [13] M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE* **7** (2008) 12–18.
- [14] N.G. Mortensen, J.C. Hansen, J. Badger *et al.* *Wind Atlas For Egypt*. NREA (2005).
- [15] G. Heal. The economics of renewable energy. *National Bureau Economic Res.* (2009).
- [16] M. Juenger, T.M. Liebling, D. Naddef, G. Nemhauser, W.R. Pulleyblank, W.R. Reinelt, G. Rinaldi and G. Wolsey, *50 Years of Integer Programming 1958-2008*. Springer (2010).
- [17] H. Kellerer, U. Pferschy and D. Pisinger, *Knapsack Problems*. Springer, Berlin (2004).
- [18] R. King, H. Rughooputh and K. Deb, Evolutionary multi-objective environmental/economic dispatch: Stochastic versus deterministic approaches. In *Evolutionary Multi-Criterion Optimization*. Springer (2005) 677–691.
- [19] J.M. Loiter and Norberg-Bohm V. Technology policy ad renewable energy:public roles in the development of new energy technologies. *Energy Policy* **27** (1999) 85–97.
- [20] S. Martello and P. Toth, *Knapsack problems: algorithms and computer implementations* (1990).
- [21] T. Nakata, K. Kubo and A. Lamont. Design for renewable energy systems with application to rural areas in japan. *Energy Policy* **33** (2005) 209–219.
- [22] New and Renewable Energy Authority. *Egyptian Solar Radiation Atlas*. NREA (1998).
- [23] S.D. Pohekar and Ramachandran M. Application of multi-criteria decision making to sustainable energy planning - A Review. *Renewable and Sustainable Energy Rev.* **8** (2004) 365–381.
- [24] A. Saad, C. Gervet and S. Abdennadher, Constraint reasoning eith uncertain data using cdf-intervals, in *Proceedings of CP-AIOR'10*. Springer Verlag (2010).
- [25] A. Schriver. *Theory of Linear and Integer Programming*. John Wiley and Sons Ltd (1996).
- [26] SciPy – Scientific Tools for Python. <http://www.scipy.org/>.
- [27] M.A. Mosalam Shaltout, Solar energy variability over horizontal and inclined passive systems in egypt, in *In proceedings of PLEA'88, Energy and Buildings for temperate Climates, A Mediterranean Regional Approach*. Pergamon Press (1988).
- [28] K. Subramanyan, U.M. Diwekar and A. Goyal, Multi-objective optimization for hybrid fuel cells power system under uncertainty. *J. Power Sources* **132** (2004) 99–112.
- [29] E. Trends, Energy and resources. *The environmental information portal* (2003).
- [30] P. Van Hentenryck and L. Michel, *Constraint-Based Local Search*. The MIT Press (2005).
- [31] L. Wang and C. Singh, Environmental/economic power dispatch using a fuzzified multi-objective particle swarm optimization algorithm. *Electric Power Systems Res.* **77** (2007) 1654–1664.
- [32] N. Yorke-Smith and C. Gervet, Certainty closure: Reliable constraint reasoning with incomplete and erroneous data. *ACM Transact. Computat. Logic* **10** (2009).