Optimal Allocation of Renewable Energy Parks: A Two-stage Optimization Model *

Mohammad Atef, Carmen Gervet

Optimal Allocation of Renewable Energy Parks: A Two-stage Optimization Model *

Mohammad Atef[1] **Carmen Gervet**[1]

[1] German University in Cairo, Tagamoa El-Khamis, New Cairo city 11835, Egypt.
{mohammad.atef,carmen.gervet}@guc.edu.eg

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 ILP.

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 [2, 3]. 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, anticipated renewable engineering cost reductions [13]. However, more is needed as highlighted in [11], 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 [2]. Clearly electricity from renewable energy remains generally more expensive than 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" [5]. Today wind energy is the least expensive option but requires more maintenance, and is space consuming compared to photo-voltaic solar panels which however, are currently more expensive. Note that the forecasts in price reduction are promising though, as shown in Figure 1 [2]. This indicates that taking into account forecast measurements is a strong element of effective decision making.

Based on existing forecast studies, and each coun-

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

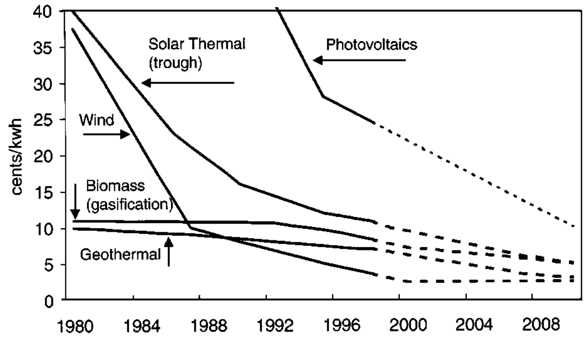

FIGURE 1 – RE costs forecast

try renewable resources, which REs or portfolio of RE should a given nation invest in ? How much should be invested now, or in 15 yeas time ? These are questions at the heart of the "economics of RE" for which our decision support tool aims to seek an answer. We seek the best trade-off cost/return on investment by taking into account physical installation constraints as well as energy requirements and costs.

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 fuel as shown in the latest earth-trends survey [21]. Egypt enjoys excellent wind and solar resources and there are tremendous potentials for investment towards local consumption and even export. Research and onsite-projects are being carried out with a growing trend.

A short-term government plan is to meet 20% of Egypt electricity demand by 2020 using RE sources. While wind farms installation is currently cheaper than solar panels of 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 [6]. Egypt can become a strong player in wind and solar energy. However a specific strategy is still to be determined, together with investments.

The optimization problem we address is defined as follows. Given the country of Egypt with its data and constraints : 1) Egypt map of populated areas, 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. 10-20 years time), such that 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 paper, we adopt 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. We consider 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 Egypt's expected electricity demand for the year 2020. To our knowledge this concept was not been considered with respect to renewable energy park placement problems.

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. Sections 4, 5 and 6 describe respectively the different models approaches we investigated. The implementation and experimental results are respectively given in Sections 7 and 8. Finally, a summary of the contribution of this work is given in Section 9 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. 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, maximizing self-production of energy, in addition to the costs factor.

In [17], 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. [15] 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, [12, 20, 24] 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 for the technical aspects that must be taken into account together with the economical ones.

[1] is the most recent and closest work to ours, where the goal of the decision support tool was to increase renewable energy parks, 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 different techniques we will be evaluating, namely dynamic programming, constrained local search and Integer linear programming.

Dynamic Programming (DP). DP is a technique occasionally used for solving optimization problems [4]. 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 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 [14]. 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 [22]. 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). This will be the case in our study.

4 Optimization Models

We now explore the approaches we considered for 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*.

4.1 The grid model

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

Input data

$$C_i = \textit{Cost of park } i \tag{1}$$
$$M_i = \textit{Maximum area of park } i \tag{2}$$
$$(x_i, y_i) = \textit{Coordinates of park } i \tag{3}$$
$$G_{ij} = \textit{Watts/m}^2 \textit{ produced by park } i \textit{ in month } j \tag{4}$$
$$D_j = \textit{20\% Electricity demand in Egypt,month } j\textit{, 2010} \tag{5}$$

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.

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

$$\mathbf{B_i} = \begin{cases} 1 & \textit{if park } i \textit{ is chosen} \\ 0 & \textit{otherwise} \end{cases} \tag{6}$$

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 :

$$\sum_{i=1}^{n} G_{ij} \cdot \mathbf{B_i} \geq D_j \quad \mathbf{B_i} \in \{0,1\} \quad \forall j \in \{1,..,12\} \tag{7}$$

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

$$\textit{minimize} \quad \sum_{i=1}^{n} \mathbf{B_i} \cdot C_i \tag{8}$$

4.2 Multi-dimensional Knapsack equivalence

We now 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 c. 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 :

$$\text{Maximize} \sum_{i=1}^{n} P_i \cdot \mathbf{X_i} \tag{9}$$
$$\text{subject to} \sum_{i=1}^{n} W_i \cdot \mathbf{X_i} \leq Q, \quad \mathbf{X_i} \in \{0,1\} \tag{10}$$

where $\mathbf{X_i} = 1$ if and only if the i^{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 :

$$\text{Maximize} \sum_{i=1}^{n} P_i \cdot \mathbf{X_i} \tag{11}$$
$$\text{subject to} \sum_{i=1}^{n} W_{ij} \cdot \mathbf{X_i} \leq Q_j, \mathbf{X_i} \in \{0,1\}, \forall j \in \{1,..,m\} \tag{12}$$

where W_{ij} is the weight consumed by the i^{th} element in the j^{th} dimension, and Q_j is the upper limit of the j^{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 (12). An optimal solution is one that maximizes (11).

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.

5 Algorithms

We now 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. The third and most successful approach uses Integer Linear Programming.

5.1 Dynamic Programming Algorithm

The most efficient algorithm for solving the KP and MKP is based on DP [14]. We 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 kWh, we can divide the problem into two independent subproblems. In the first subproblem, the n^{th} location is included in the solution, the cost associated with the n^{th} is subtracted in the new subproblem and the cost is added. In the second subproblem the n^{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 :

$$(n, C, G, d) \tag{13}$$

where n is the number of locations, c is the rate of electricity required in kWh, G_n is the production gain of kWh provided by the n^{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 :

$$F(n,d) = \begin{cases} 0 \text{ if } d \leq 0 \\ \infty \text{ if } n = 0 \text{ and } d > 0 \\ \min \begin{pmatrix} F(n-1, d) \\ C_n + F(n-1, d - G_n) \end{pmatrix} otherwise \end{cases} \tag{14}$$

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^{th} location and the other that excludes it.

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 $\{1, .., 12\}$), 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 $d = \max_{j=1}^{m} \mathrm{D}_j$.

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 one 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 (14), which prunes the computation once the demand has been satisfied. Computational results are presented in the implementation section.

Preprocessing : adding a sorting heuristic A preprocessing step to this approach is to sort the locations before running the DP algorithm. The sorting rule is defined as follows :

$$Location_a < Location_b \iff \sum_{j=1}^{m} G_{aj} < \sum_{i=1}^{m} G_{bj}$$

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 8, 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.

5.2 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. [9]. 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 [9] that are highlighted here.

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 $\textsc{Neighborhood}(\mathrm{Y}) = \{[\underline{2}, \underline{1}, 3, 4], [\underline{3}, 2, \underline{1}, 4], [\underline{4}, 2, 3, \underline{1}], [1, \underline{3}, \underline{2}, 4], [1, \underline{4}, 3, \underline{2}], [1, 2, \underline{4}, \underline{3}]\}$. However, $[3, 1, 2, 4]$ and $[4, 3, 2, 1] \notin \textsc{Neighborhood}(\mathrm{Y})$ since they each contain two swaps.

Algorithm 5.1 GREEDY

Require: G, D, Y
  $\mathbf{E} \leftarrow \{\mathbf{E}_i = 1, \forall 1 \leq 1 \leq n\}$ {solution vector with all parks selected}
  $V \leftarrow [V_1, .., V_m]$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 $E - G_{Y_i} \geq D$ **then**
      $E \leftarrow E - G_{Y_i}$
      $\mathbf{V}_{Y_i} \leftarrow 0$
    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 (GREEDY(G,D,Y)), compute the set of neighboring vectors, determine if a neighbor point can lead to a better solution cost (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 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 E causes the total demand to be unsatisfied, the algorithm does not remove it, otherwise it is removed.

Algorithm 5.2 Local Search

Require: G, C, D
  Y $\leftarrow [Y_1, Y_2, ..., Y_n]$ {*random permutation*}
  $\mathbf{X} \leftarrow$ GREEDY(G, D, Y)
  loop
    Y' $\leftarrow$ FINDBEST(Y) {*finds the best permutation in the neighborhood of Y which is better than Y*}
    if Y' does not exist **then**
      return $\mathbf{X}$
    end if
    $\mathbf{X} \leftarrow$ GREEDY(G, D, Y') {*binary solution from the greedy algorithm (5.1)*}
  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 neighbor 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 [1].

Complexity The highest cost of Algorithm 5.2 is FINDBEST, which finds the best next solution, better than the current one from within the neighborhood. The time complexity of FINDBEST is polynomial in the number of locations.

$$O(n^3 \cdot m) \tag{15}$$

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 (5.2) 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.

1. This is called local search with memory, which resembles the operation of tabu search.

5.3 Integer Linear Approach

Finally we tried a third approach which proved the most effective in terms of both, solution quality and efficiency. The constraints in Equation (7) are linear constraints ensuring that the demand for each month is satisfied by the chosen potential locations.This grid model can be solved using the Simplex algorithm to find a solution which might violate the Integrality constraint on the variables, and is then combined with a branch and bound algorithm to search for integer values. Solving the grid model with an ILP solver proved to be the best approach, and we will use it for the two-stage optimization model.

6 Two-stage optimization model

We now extend the grid model (Section 4), 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).

	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**

TABLE 1 – All possible scenarios of building a and b

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 [2]. 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 :

$$C_i = \textit{Cost of park i} \tag{16}$$

$$\underline{FC_i} = \underline{\textit{Future cost of the } i^{th} \textit{ location}} \tag{17}$$

$$M_i = \textit{Maximum area of park i} \tag{18}$$

$$(x_i, y_i) = \textit{Coordinates of park i} \tag{19}$$

$$G_{ij} = \textit{Watts}/m^2 \textit{ produced by park i in month j} \tag{20}$$

$$D_j = \textit{Electricity demand in 2010,month j} \tag{21}$$

$$\underline{r} = \underline{\textit{Annual growth rate of electricity demand}} \tag{22}$$

$$\underline{FD_j} = \underline{D_j \times r \textit{ future demand, month j, 2020}} \tag{23}$$

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

$$\mathbf{X_i} = \begin{cases} 1 & \textit{The } i^{th} \textit{ location is to be built in the present} \\ 0 & \textit{otherwise} \end{cases} \tag{24}$$

$$\underline{\mathbf{FX_i}} = \begin{cases} 1 & \textit{The } i^{th} \textit{ location is to be built in the future} \\ 0 & \textit{otherwise} \end{cases} \tag{25}$$

Constraints The integer linear program has the following linear constraints :

$$\sum_{i=1}^{n} G_{ij} \cdot \mathbf{X_i} \geq D_j, \ \forall j \in \{1, .., m\} \tag{26}$$

$$\underline{\sum_{i=1}^{n} G_{ij} \cdot (\mathbf{X_i} + \mathbf{FX_i}) \geq FD_j, \ \forall j \in \{1, .., m\}} \tag{27}$$

$$\underline{0 \leq \mathbf{X_i} + \mathbf{FX_i} \leq 1, \ \forall i \in \{1, .., n\}} \tag{28}$$

$$\mathbf{X_i} \in \{0, 1\}, \quad \mathbf{FX_i} \in \{0, 1\}, \ \forall i \in \{1, .., n\} \tag{29}$$

Constraints 26 ensure that the locations chosen to be built in the present satisfy the current demand, while constraints 27 ensure that all chosen locations built in the present and the future satisfy the future demand as well. Constraints 28 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 :

$$Cost = \sum_{i=1}^{n} \mathbf{X_i} \cdot C_i + \sum_{i=1}^{n} \mathbf{FX_i} \cdot FC_i \tag{30}$$

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.

7 Implementation

We now discuss the implementation of the different algorithms used and our prototype. 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 (14) as well as the constrained local search one (5.2), were implemented in `C++`. The DP algorithm uses top-down DP. The ECL^{i}PS^e [23] platform was chosen to implement the ILP model thanks to its high level of abstraction and hybridization libraries with the `eplex` solver.

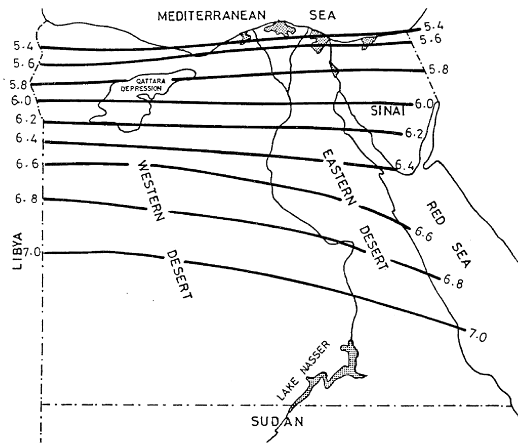

FIGURE 2 – Solar radiation map for March

Data pre-processing We purchased from a governmental body the wind and solar solar atlas of Egypt [16, 8]. 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^2 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 is shown in Figure 2. The information in these maps was extracted using `Python`'s image processing library `SciPy` [19]. 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 [7]. 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.

Dataset Generation A dataset is composed of n points chosen and marked on a map like Figure 2. The gain of these locations is derived from the gain matrix G for the dataset. The costs for these locations is predefined according to the type of park considered (wind, solar).

Graphical User Interface The GUI module was implemented using `Java`. In order to enable the user to freely select locations from the map of Egypt, the OpenStreetMap [10] 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 Evaluation

We first present the results of evaluating the scalability of the different algorithms we implemented : 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.

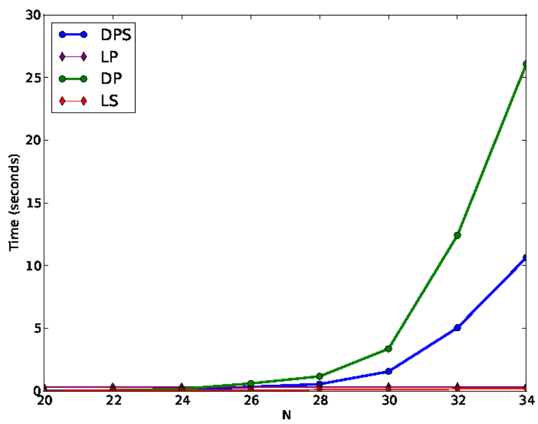

FIGURE 3 – Runtime of DP, DPS, ILP, and LS approaches for $20 \leq n \leq 34$

In Figure 3 we show the CPU running times of all approaches for a group of datasets where $20 \leq n \leq 34$. 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.

Now we show the performance of the LS and ILP algorithms for larger values of n, $40 \leq n \leq 220$, see Figure 4. 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 yet proves to be extremely fast and robust in solving all instances.

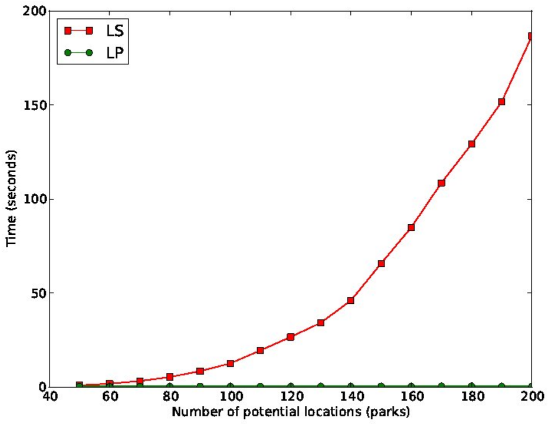

FIGURE 4 – Runtime of ILP and LS approaches for $40 \leq n \leq 220$

Forecast ILP Model The solution was broken into two stages. First the simple gridmodel described in Section 4 is solved. 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.

We show that adding this constraint and solving the two-stage problem with the constrained new objective function (6), even though it requires a second run of the Cplex solver, it greatly speeds up the runtime of the entire algorithm. In other words, solving the forecast model without the upper bound constraint on the cost is less efficient than solving the grid model, acquiring the upper bound, then solving the forecast model, as shown in Figure 5. The testing was done for larger number of potential locations, $50 \leq N \leq 250$ given its efficiency.

Solution Quality We now illustrate the impact of reasoning with forecast demand and cost values in terms of the solution quality and difference in final

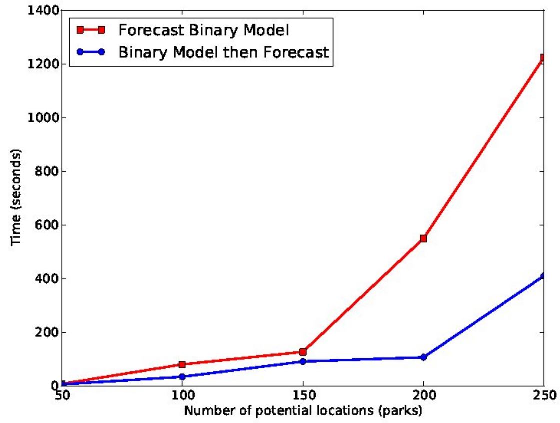

FIGURE 5 – One stage versus two-stage execution with forecast data

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 6, the optimal set is shown when only current data is considered. While in Figure 7, it is shown how the optimal solution differs when we account for forecast costs and demands. We can see that solar park 8 and 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.

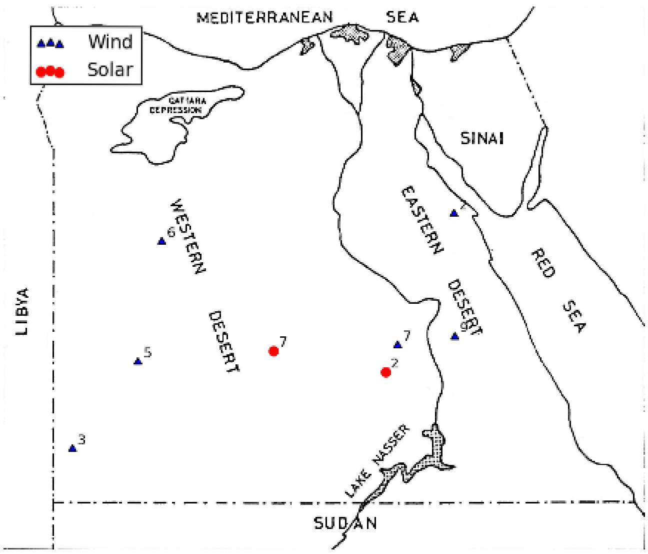

FIGURE 6 – Chosen locations using grid model

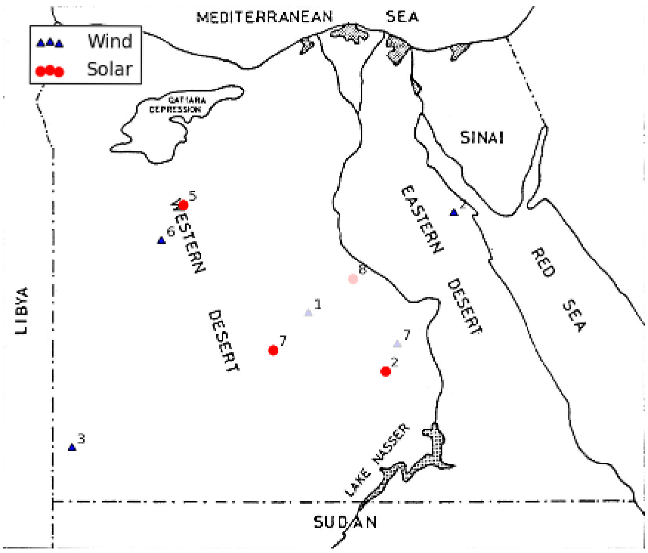

FIGURE 7 – Chosen locations with two-stage model

9 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. Future work includes the integration of our prototype into a full software made available to governmental investors and domain experts. From a research point of view, we also aim at fine investigating different models to represent the forecast data such as CDF-intervals [18] and probabilistic approaches.

Références

[1] 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.

[2] S.R. Bull. Renewable energy today and tomorrow. *Proceedings of the IEEE*, 89(8) :1216–1226, 2001.

[3] S.R. Bull. The integration of renewables. 2004.

[4] S. Dreyfus. Richard bellman on the birth of dynamic programming. *Operations Research*, pages 48–51, 2002.

[5] Energy efficiency ad Renewable energy. www.eere.energy.gov/wind.

[6] M.N. El-Kordy, M.A. Badr, Abed K.A., and Ibrahim S.M.A. Economical evaluation of electricity generation considering externalities. *Renewable Energy*, 25 :317–328.

[7] Saher El-Neklawy. Extracting solar radiation data from egyptian solar atlas (to appear). Technical report, German University in Cairo, August 2011.

[8] J. C. Hansen et al. *Wind Atlas For Egypt.* NREA, 2005.

[9] D. Ghosh, N. Chakravarti, SS Fatima, M. Wooldridge, and NR Jennings. A competitive local search heuristic for the subset sum problem. *Computers and Operations Research*, 26(3) :271–280, 1999.

[10] M. Haklay and P. Weber. Openstreetmap : User-generated street maps. *Pervasive Computing, IEEE*, 7(4) :12–18, 2008.

[11] G. Heal. The economics of renewable energy. *National Bureau of Economic Research*, 2009.

[12] R. King, H. Rughooputh, and K. Deb. Evolutionary multi-objective environmental/economic dispatch : Stochastic versus deterministic approaches. In *Evolutionary Multi-Criterion Optimization*, pages 677–691. Springer, 2005.

[13] J.M. Loiter and Norberg-Bohm V. Technology policy ad renewable energy :public roles in the development of new energy technologies. *Energy Policy*, 27 :85–97, 1999.

[14] Silvano Martello and Paolo Toth. Knapsack problems : algorithms and computer implementations. 1990.

[15] T. Nakata, K. Kubo, and A. Lamont. Design for renewable energy systems with application to rural areas in japan. *Energy Policy*, 33(2) :209–219, 2005.

[16] New and Renewable Energy Authority. *Egyptian Solar Radiation Atlas.* NREA, 1998.

[17] S.D. Pohekar and Ramachandran M. Application of multi-criteria decision making to sustainable energy planning - A Review. *Renewable and Sustainable Energy Reviews*, 8(4) :365–381, 2004.

[18] A. Saad, C. Gervet, and S. Abdelnnadher. Constraint Reasoning with Uncertain Data using CDF-Intervals. *In proceedings of CP-AIOR'10*, pages 292–306, 2010.

[19] SciPy – Scientific Tools for Python. http ://www.scipy.org/.

[20] K. Subramanyan, U.M. Diwekar, and A. Goyal. Multi-objective optimization for hybrid fuel cells power system under uncertainty. *Journal of power sources*, 132(1-2) :99–112, 2004.

[21] Earth Trends. Energy and resources. *The environmental information portal*, 2003.

[22] P. Van Hentenryck and L. Michel. *Constraint-Based Local Search.* The MIT Press, 2005.

[23] M. Wallace, S. Novello, and J. Schimpf. Eclipse : A platform for constraint logic programming. *ICL Systems Journal*, 12(1) :159–200, 1997.

[24] L. Wang and C. Singh. Environmental/economic power dispatch using a fuzzified multi-objective particle swarm optimization algorithm. *Electric Power Systems Research*, 77(12) :1654–1664, 2007.