Wind Farm Optimization

S. Donovan
Department of Engineering Science
University of Auckland
New Zealand
sdon032@ec.auckland.ac.nz

Abstract

This paper will formulate integer programs to determine the optimal positions of wind turbines within a wind farm.

The formulations were based on variations of the vertex packing problem. Three formulations are presented, which seek to maximize power generated in accordance with constraints based on the number of turbines, turbine proximity, and turbine interference. These were in the form of budget, clique, and edge constraints.

Results were promising, with turbines exhibiting a tendency to concentrate in areas of high elevation and avoid situations where downstream interference would be significant.

1. Development of the integer programs

Three (mixed) integer programming models are presented. The first two integer program models are vertex packing problems, while the third MIP model is a Generalized Vertex Packing Problem (GVP) [1]. The GVP problem was introduced by Hanif D. Sherali and J. Cole Smith [2].

In these formulations, G = (V, E) denotes a graph with vertices V and edges $E \subseteq V \times V$. The set E is set of vertex pairs between which there exists some relationship. In our case, the vertices V correspond to the locations where turbines can be positioned, and the edges E represent relationships between the vertices, such as turbine proximity and interference.

An appreciation of the relationship between the physical domain and the graph on which the (mixed) integer programs are based is crucial to understanding the material that follows.

The graph is based on an orthogonal grid that is superimposed onto the physical topography. The intersection points of this grid represent the vertices in our graph. The vertex packing problem will thus involve selecting the combination of vertices, or grid points, which generates the most power.

2. Modeling turbine proximity

The first integer program formulation enforced a minimal separation distance between turbines to ensure the blades did not physically clash with one another. The term proximity shall define the area immediately surrounding a turbine in which no other turbine can be built. The grid points that lie within this area are a function of the turbine radius and the physical distance between the intersection points in our grid.

Figure 2.1 demonstrates that a turbine centered on the solid vertex will eliminate the surrounding vertices as potential locations. That is, the vertices connected to the solid vertex by an edge are too close to accommodate another turbine. Those vertices that are not connected to the solid vertex do not impinge on the space required by this turbine.

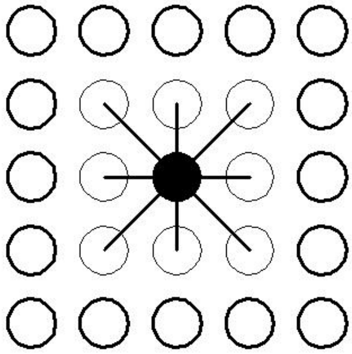

Figure 2.1: Vertex proximity constraint.

To model this proximity requirement we construct the graph G by considering each vertex in turn, and placing an edge between this vertex *i* and any vertex *j*, where the position occupied by *j* violates the area required by a turbine positioned at *i*. For example, in Figure 2.1 an edge would exist between the solid vertex and every surrounding vertex, as shown by the lines. This constraint can be formulated mathematically as:

$$x_i + x_j \leq 1, \forall (i, j) \in E$$

An edge constraint of this form will exist between every vertex in our graph and any other vertex that lies within the required separation distance. In the physical model, this corresponds to a pair of grid points existing too close for a turbine to be located at both positions.

The above "weak edge" formulation can be improved by considering a larger subset of vertices affected by turbine proximity. The structure of the edges on *G*, as well as the relationship between four neighboring vertices Q, is shown in Figure 2.2.

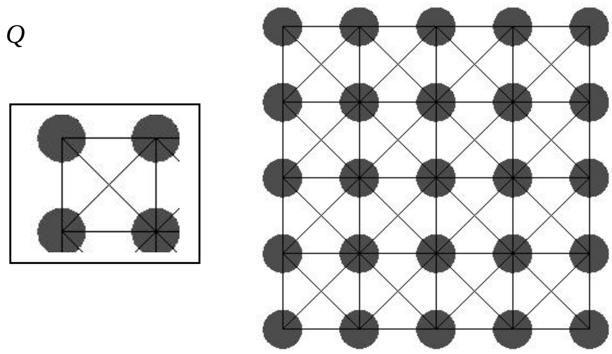

Figure 2.2: Clique structure.

A maximal clique Q is defined as a maximal subset (with respect to cardinality) of vertices whereby all vertices in the subset are connected by an edge to all the other vertices in the subset. For the subset Q shown in Figure 2.2, the clique would involve all four vertices connected by an edge, with the sum of the turbines constructed in that subset constrained to be less than or equal to one. In general terms, the cardinality of a clique will be a function of the turbine radius and distance between the grid points in the x and y direction. Let K denote the set of all maximal cliques in our graph G. Each maximal clique $Q \in K$ is a subset of V.

Let W_v denote the power value associated with a vertex v. Let $x_v=1$ denote a turbine positioned at vertex v, and $x_v=0$ otherwise. A budget constraint restricts the maximum number of turbines to be built in the wind farm to be less than or equal to k.

The integer program can now be formulated as:

$$
\begin{aligned}
&\text{Maximize} && \sum_{v \in V} W_v x_v \\
&\text{Subject to} && \sum_{v \in V} x_v \le k \\
& && \sum_{v \in Q} x_v \le 1, \quad \forall Q \in K \\
& && x_v \in \{0,1\}, \quad v \in V
\end{aligned}
$$

This formulation shall be referred to as IP1.

IP1 led to dense clusters of turbines in areas of high power resource. This stems from the fact that the only constraint on turbine location was the proximity constraint defined by the radius of the turbine blades. To this end, IP1 ensures that turbine proximity is not violated. It does not, however, reflect the influence of interference on the amount of power generated at downstream locations.

An example of a 50 turbine farm optimized using IP1 is shown in Figure 2.3, where the turbine locations are indicated by the dots. Recall this has been optimized for wind flow from the west.

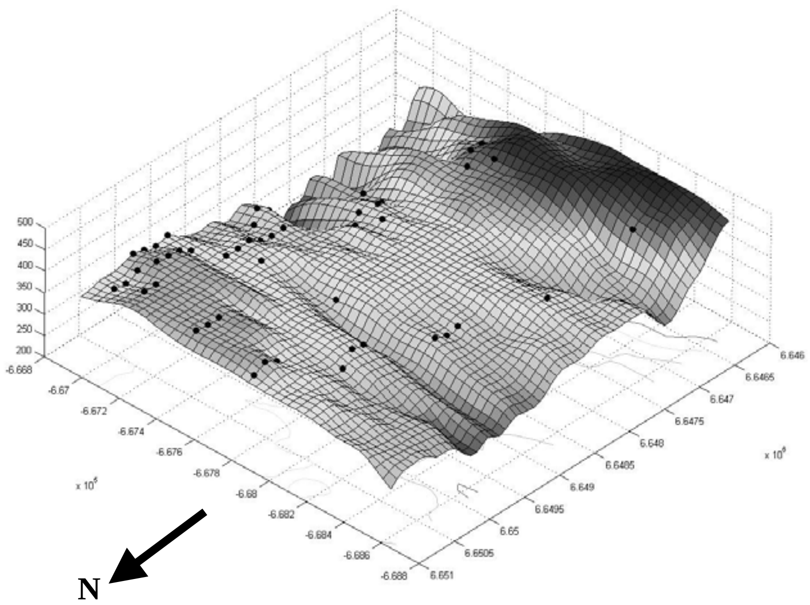

Figure 2.3: IP1 results (50 turbines).

The solution to IP1 has the following features:

1. Turbines are concentrated in areas of high power resource.
2. The proximity constraint prevents turbines from occupying adjacent vertices.

3. Modeling turbine interference

Interference between turbines was not taken into account in the formulation of IP1. The industry standard recommended separation distance will now provide the basis for an integer program formulation that accounts for turbine interference. The distance required between turbines is taken to be 7 turbine diameters when aligned in the predominant direction of flow, and 3 turbine diameters in the other direction [3]. Figure 3.1 demonstrates these separation distances when a turbine is centered at the shaded vertex.

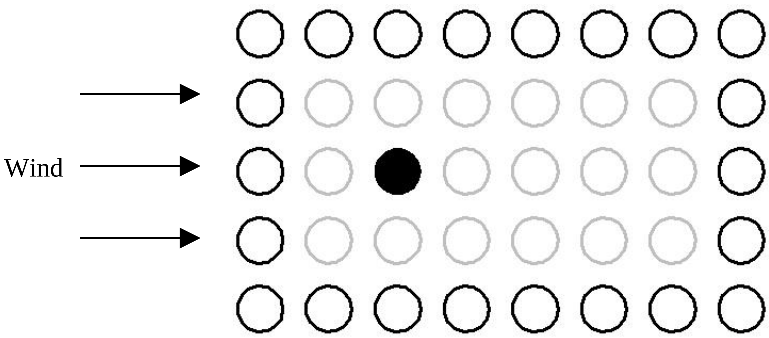

Figure 3.1: Fixed separation distance.

Figure 3.1 shows that a turbine centered at the solid vertex will influence the immediate surrounding vertices as well as vertices further downstream. Vertices outlined with a dark line are unaffected. This model of interference will, therefore, increase the size of the cliques to reflect both turbine proximity and turbine interference. This formulation shall be referred to as IP2.

The results from IP2 varied significantly from IP1, as shown in Figure 3.1.

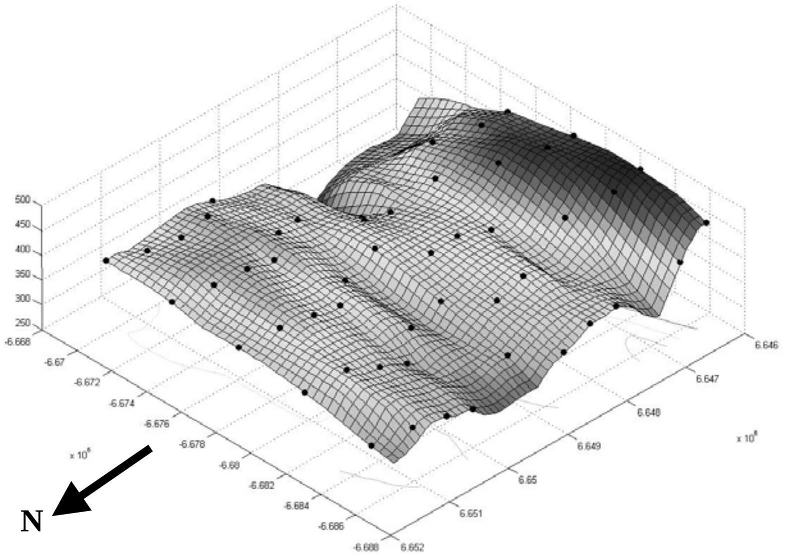

Figure 3.2: IP2 results (50 turbines).

The solution to IP2 has the following unique features:

1. The larger cliques prevent turbines from clustering
2. Turbines are aligned perpendicular to the main direction of flow, reflecting the smaller separation distance imposed in this direction.

IP2 is an improvement over IP1 because turbine interference is taken into account.

4. A better model for turbine interference

While IP2 is an improvement over IP1, the imposition of an arbitrary separation distance between turbines is a blunt approach to turbine interference. Instead, it would be better to position turbines according to the net power gain, which is defined as the amount of power generated less the magnitude of interference.

This separates the area surrounding each turbine into two distinct measures, which reflect:

1. Proximity
2. Interference

These measures are shown in Figure 4.1.

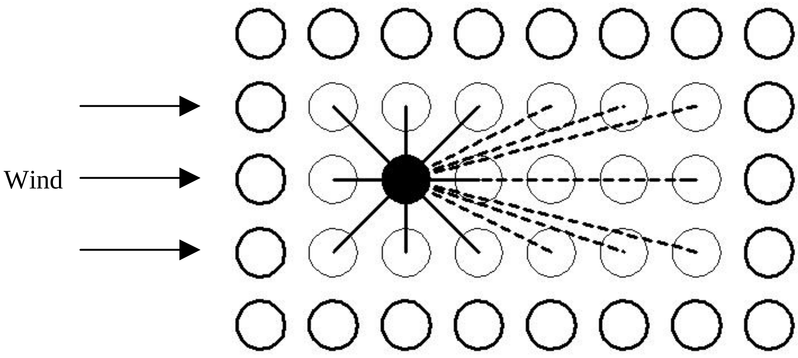

Figure 4.1: Proximity and interference effects.

Figure 4.1 shows that a turbine positioned at the solid vertex will physically obstruct the surrounding vertices, to which it is connected by a solid line. As a result, the set of all cliques K is identical to the proximity constraints formulated in IP1.

However, this model distinguishes itself by introducing edge based on the interference between vertices. The graph G considers each vertex in turn, and places an edge between this vertex *u* and any vertex *v* that experiences interference above a certain magnitude. In Figure 4.1, for example, an edge would exist between the solid vertex and every vertex that is connected to it via a dashed line.

Let E_I denote the set of edges between all vertices *u* and *v* that interfere with each other. Variable $z_{uv}=1$ if a turbine is positioned at *u* and *v*, and $z_{uv}=0$ otherwise. This can be formulated mathematically as:

$$x_u + x_v - 1 \le z_{uv}$$

Let I_{uv} denote the magnitude of the power loss caused by interference between vertices *u* and *v*. The methods used to determine *I* are outside the scope of this paper. If $z_{uv}=1$, which denotes that a turbine is positioned at both x_u and x_v, then the expected value of the power generated will decrease by the amount I_{uv}.

The mixed integer program formulation, which shall be referred to as MIP1, then becomes:

$$\begin{aligned}
\text{Maximize} \quad & \sum_{v \in V} W_v x_v - I_{uv} z_{uv} \\
\text{Subject to} \quad & \sum_{v \in V} x_v \le k \\
& \sum_{v \in Q} x_v \le 1 \\
& x_u + x_v - 1 \le z_{uv}, (u, v) \in E_I \\
& x_v \in \{0,1\}, v \in V, Q \in \theta, z \ge 0
\end{aligned}$$

The decision variables *z* are not constrained to be binary because the formulation enforces them to take binary values of 0 or 1. This follows from *I* being strictly positive. A variable z_{uv} takes a value of 0 unless the corresponding interference constraint forces it to assume a value greater than or equal to 1. In this event, the deleterious impact of *I* on the objective function causes z_{uv} to assume the smallest value possible, which is 1.

The results generated using MIP1 are shown in Figure 4.2.

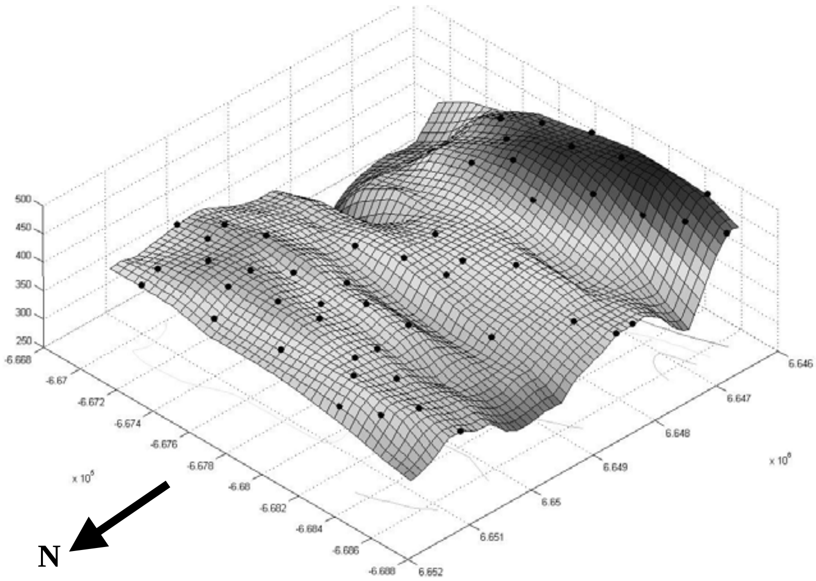

Figure 4.2: MIP1 results (50 turbines).

The major difference between IP2 and MIP1 is in the ability of the latter to evaluate the balance between power generation and interference losses. This means that in situations of high power resource, MIP1 is willing to accept interference if the net power generated will be more than what would result from another position.

Therefore, the MIP1 formulation is distinct from IP1 and IP2 in that it strictly enforces turbine proximity, while accounting for turbine interference using edges. MIP1 can evaluate the net benefit of turbine interference in positions of high power resource, and position turbines accordingly.

5. Analysis of results

All of the aforementioned formulations were modeled in AMPL and solved using CPLEX version 6.6.0 with default settings. AMPL is a text based algebraic modeling language used to program optimization problems. CPLEX is a commercial solver for mixed integer problems.

The difference between the expected power outputs for IP2 and MIP1 reflects the value of compromising on the separation distance between turbines in areas of high power resource. MIP1 outperformed IP2, particularly as the maximum number of turbines k was increased. This trend is shown in Figure 5.1.

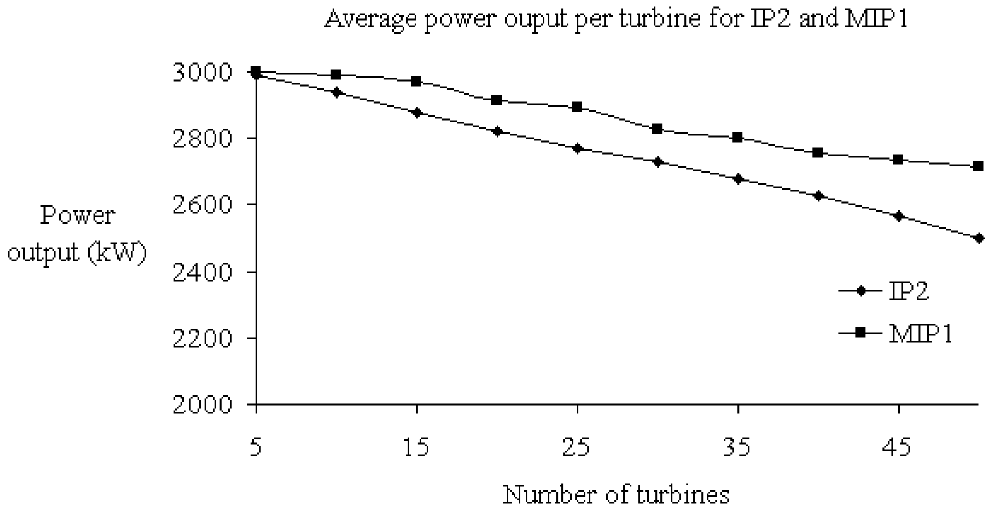

Figure 5.1: Comparison of average power output from IP2 and MIP1.

Figure 5.1 demonstrates that, for wind farms involving more than 45 turbines, the optimal configuration determined by MIP1 would generate approximately 10 percent more power than IP2. This is a significant gain.

6. Performance of MIP1

General vertex packing problems, which contain vertex packing problems as a specific case, are NP-hard. It is extremely unlikely that there exists an efficient algorithm to solve these problems. Efficient means that the algorithm will run in polynomial time.

This section will assess the performance of MIP1. The number of decision variables involved in MIP1, as determined by the number of vertices and interference edges, is a function of the number of grid points and the methods used to determine interference.

Graphs of the problem size against number of variables and number of non zeros are shown in Figures 6.1 and 6.2.

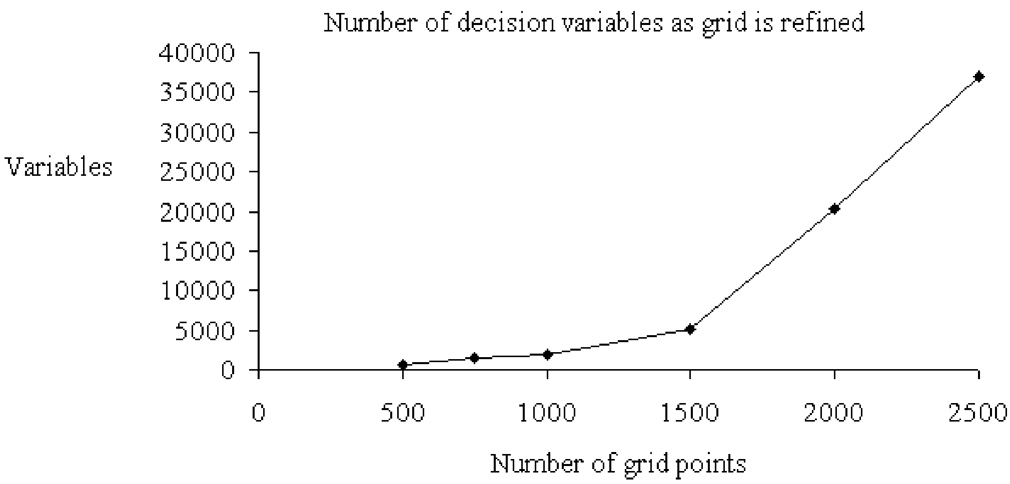

Figure 6.1: Number of variables versus problem size.

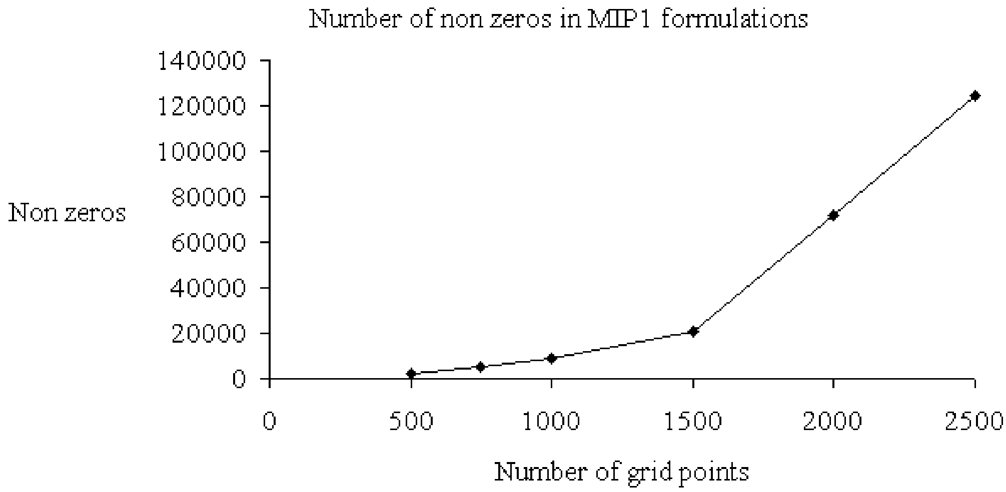

Figure 6.2: Number of non zeros versus problem size.

Moreover, the maximum number of turbines in the farm, *k*, also had a significant impact on the time taken to solve to optimality. For a sample problem involving 2400 grid points, only configurations involving less than 10 turbines could be solved to optimality in less than one hour. This was on a machine operating Windows 2000 with a 3.00MHz Pentium IV processor and 1 gigabyte of RAM.

The gap between the value of the best feasible solution and the best bound found after one hour is plotted in Figure 6.3.

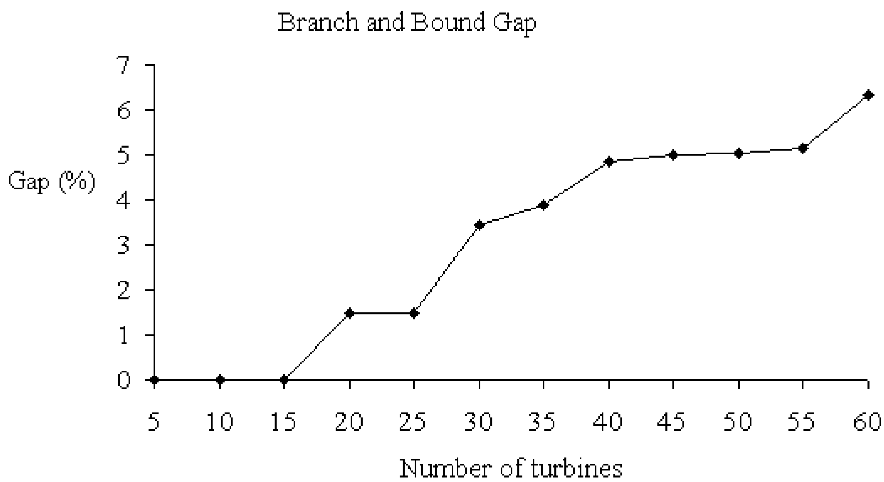

Figure 6.3: Branch and bound gap.

7. Conclusions and future work

This paper has outlined methods for the optimization of turbine locations within a wind farm. Several integer program formulations were presented. The final formulation was designed to maximize the total power generated while observing turbine size and turbine interference. Constraint formulation was based on cliques and edges derived from the underlying vertex graph.

The integer programs could be solved close to optimality in acceptable time. Results were consistent with expectations, with turbines exhibiting a preference for areas of high elevation. The MIP used in this project performed well, even when confronted with relatively large problems. However, the potential exists for the model to be extended beyond its current form.

A potentially important constraint would exclude areas of unduly steep terrain from being selected as a turbine location. This could be incorporated by using information on topographical gradients.

Another interesting constraint would involve relating the total distance between turbines to some construction cost that reflects, for example, the length of trenching required. This would involve another "interference" type sub graph, where cost was linear in variables defined over vertex pairs. More complex interference shapes could be modeled using a set packing formulation.

The budget constraint *k* could be replaced by a measure of productivity, which controlled the maximum number of turbines that are built. The MIP would then construct as many turbines as possible, while satisfying some minimum output for each turbine. This minimum output could be determined by considering the desired payback period for the wind farm investment.

This means that for every turbine location *i*, the amount of power generated less the interference experienced must exceed some critical value, denoted by *P*. This constraint is formulated mathematically as:

$$w_i x_i - \sum_{(i,j)\in E_I} I_{ji} z_{ij} \geq P \quad \forall i \in V$$

There are also opportunities for the application of heuristic algorithms to work towards improved solutions. These may be particularly useful for problems involving nonlinear constraints, such as noise and line of sight, or for improving on a non optimal solution generated by the branch and bound process. The greater complexity of these constraints may well be suited to heuristic, rather than exact, solution methods.

8. References

[1] *A polyhedral study of the generalized vertex packing problem*, H. D. Sherali and J. C. Smith, Math. Programming, to appear.

[2] *A class of web-facets for the generalized vertex packing problem*, H. D. Sherali and J. Cole Smith, in 'Discrete Applied Mathematics', Vol. 146 (2005), pp. 273 – 286.

[3] *Fact Sheet 5: Siting*, New Zealand Wind Energy Association, The Cost of Wind Energy, www.windenergy.org.nz/FactSheets/Siting5, accessed 13/4/05, last updated 13/6/05.