Outline

- Maximum Independent set Problem
- Traveling Salesmen problem
- Metric Traveling salesmen problem
Maximum Independent Set problem

Even though two problems are related like Vertex cover and Max-Independent set problem, having an approximation algorithm for one does not mean that we have one for another too. Now we are going to prove this fact.

Let $G=(V,E)$ be an undirected graph
- $|V|=n$ total number of vertices in the graph
- $|E|=m$ total number of edges

Let $V_{cover}$ be the vertex cover of $G$
- $V_{cover} \subseteq V$
- $V - V_{cover}$ is a corresponding independent set

- $V_{approx}$ -> Approximate vertex cover.
- $V_{opt}$ -> Optimal vertex cover
- $I_{approx}$ -> Approximate Independent set
- $I_{opt}$ -> Optimal Independent set

$V_{approx} \leq 2V_{opt}$  
- i.e. the vertex cover got by approximation algorithm is at max 2 times the optimal solution of vertex cover

$V - V_{approx} = I_{approx}$
- $I_{opt} = V - V_{opt}$
Now to express $I_{aprox}$ as a factor of $I_{opt}$ as in the equation 2 for vertex cover, we need a constant factor say 'C'.

$I_{aprox} \geq C \cdot I_{opt}$

we are interested in finding the constant C

Let $|V_{opt}| = x$

$|V_{aprox}| = y$

$|I_{aprox}| = z$

$|I_{opt}| = u$

Now from the equations 1, 2 & 3 we get

\[ y \leq 2x - 4 \]

\[ n - y = z \]

\[ u = n - x \]

and we know that $x \leq y \leq n$

we need the relation between $z$ and $u$

from the equation 4, 5 and 6 we get

\[ y = n - z \]

\[ x = n - u \]

\[ \frac{y}{x} = \frac{n - z}{n - u} \leq 2 \]

\[ n - z \leq 2n - 2u \]

\[ n - z \leq 2u - n \]

it is impossible to deduce this equation further to express $u$ as a factor of $z$. 
Considering a graph of 100 vertex, the fact that the relationship cannot be established between a vertex cover approximation algorithm and Max Independent set is shown below.

\[ V_{opt} - V_{aprox} = 90 = I_{opt} \] is close to approx so the \( I_{aprox} \) here is acceptable.

\[ V_{opt} - V_{aprox} = 10 \] is not near the \( I_{aprox} \)

ie. \( I_{opt} \) is a factor of 10 with respect to \( I_{aprox} \) which is not acceptable.

So no particular constant exists to connect \( I_{aprox} \) and \( I_{opt} \) and thus no relation exists.
Traveling Salesman Problem

Input:
G=(V,E) a weighted complete graph

Output:
The shortest tour which visits each vertex exactly once.

For the general problems, a constant factor approximation is impossible unless P=NP

Proof:
Assume ∃ optimization algorithm 'A' that run in polynomial time grantees a approximation factor of c>=1

now if we have an algorithm to solve the TSP problem
for a given graph G , we add all the remaining edges to make the graph complete and give a weight of '0' to the original edges and '1' to the newly added edges. And we ask the TSP solution black box whether a tour exists of cost 0.
Now the TSP solution black box finds a circuit of cost '0'
now if we had the constant $C$, say 100
$C \times \text{cost} = 100 \times 0 = 0$
so no such constant exists.

If equal to '0'
yes;
else
No;

Graph $G$

General TSP cannot be approximated by a general factor
This is a restriction version of TSP.
A very natural restriction of the TSP is to require that the distances between cities form a metric, i.e., they satisfy the triangle inequality. That is, for any 3 cities A, B and C, the distance between A and C must be at most the distance from A to B plus the distance from B to C. Most natural instances of TSP satisfy this constraint.

A simple greedy algorithm for this is, for each step chose the shortest distance of the remaining edges.

An algorithm which is efficient than this is discussed in the next page
We construct a Minimum spanning tree from the given graph, and add an extra edge between each vertex between which an edge already exists.

After adding an extra edge we go through each vertex and tour back until all the vertex are covered but while coming back we consider only the unvisited edges and draw an edge between last visited and the current unvisited vertex (doted line) which gives us the second graph above an TSP
Topt >= MST (viewing TSP as one more edge to MCST
Tprime=2 MCST (we can see this in the graph last page)
Tprime>=Tapprox
Tapprox>= 2 Topt

Note: a factor of 3/2 approximation algorithm also available
we can do this using Maximum weight machine in graph