Algoritmo específico:

Se trata de un algoritmo muy básico de orden O(n²) el cual se basa en la comparación de distancias entre todas las vacas que hay en el problema, guardando el índice de las vacas con la distancia menor y esta distancia. Una vez se hayan comparado todas las parejas de vacas posibles, tendremos los índices de las más cercanas entre sí.

Este algoritmo consta de dos bucles anidados, uno que comenzará en el principio del *array* que guarda las coordenadas de las vacas, y otro que comenzará en el elemento siguiente al que esté situado el bucle exterior.

IMPLEMENTACIÓN:

pair<int,int> vacas\_amigas(pair<double,double> \*vacas, int n) { // O(n²)

if (n > 1) { // O(n²)

pair<int,int> amigas(0,1); // O(1)

double dist = distancia(vacas[0],vacas[1]); // O(1)

for (int i = 0; i < n-1; i++) { // O(n²)

for (int j = i+1; j < n; j++) { // O(n)

if (i!=j) { // O(1)

if (distancia(vacas[i],vacas[j])<dist) { // O(1)

amigas.first=i; // O(1)

amigas.second=j; // O(1)

dist=distancia(vacas[i],vacas[j]); // O(1)

}

}

}

}

return amigas;

}

else return pair<int,int>(-1,-1); // O(1)

}

ESTUDIO TEÓRICO:

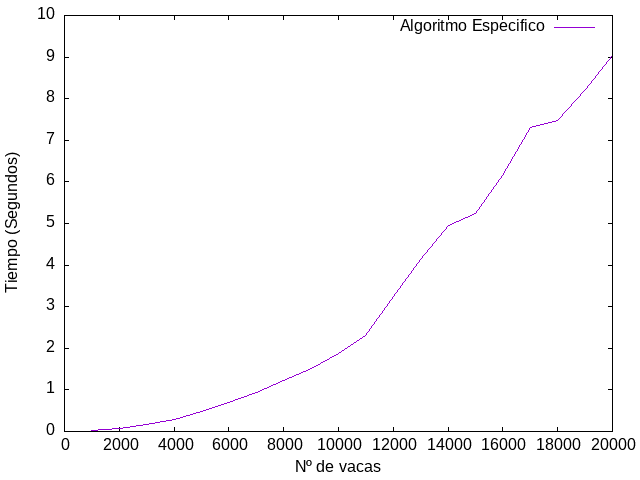
Vemos como la mayor parte del tiempo de ejecución el algoritmo se encontrará dentro del bucle interno, por lo que llamemos a este tiempo “t”. De esta forma, con n elementos en el *array* vacas, tenemos que

y vemos como obviamente . Por lo tanto, diremos que se trata de un algoritmo de orden O(n²) y que la función que mejor ajustará los tiempos de este algoritmo será un polinomio de segundo grado

ESTUDIO EMPÍRICO

Escalando el problema a distinto número de casos, comenzando con 1000 vacas y llegando hasta 20000, con saltos de 1000 en 1000, hemos estudiado los tiempos de ejecución del algoritmo. Para conseguir un resultado lo más representativo posible, cada uno de estos tiempos se ha obtenido de 5 mediciones, realizando la media entre estas. Así, hemos obtenido los siguientes resultados:

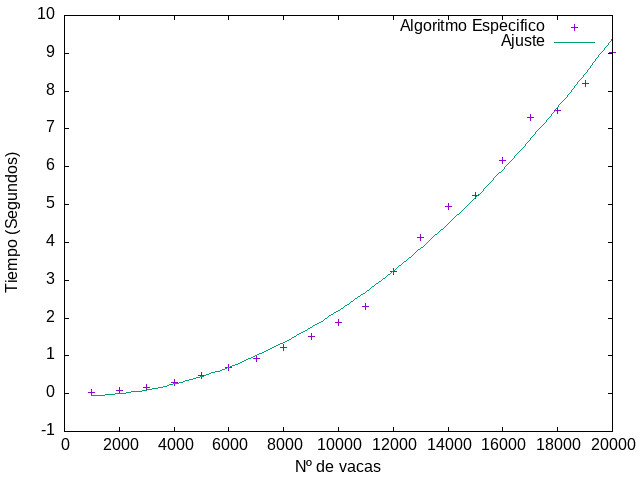
(Tabla)



ESTUDIO HÍBRIDO:

Como hemos visto en el análisis teórico, se trata de un algoritmo cuadrático, luego el mejor ajuste que podemos realizar teóricamente sobre esta gráfica será con una función polinómica de segundo grado.

Con la ayuda del software *gnuplot*, hemos realizado un ajuste por regresión mediante mínimos cuadrados con los datos obtenidos, consiguiendo



(Función con parámetros)