3.Explicación del problema de programación matemático estudiado.

En este problema se considera que necesitamos procesar *n* tareas en una máquina que solo puede procesar un trabajo a la vez. Se puede asignar cualquier tarea sin restricciones a la máquina siempre que no haya otro trabajo procesándose, es decir, no se tiene un orden obligatorio ni relaciones de precedencia. Cada tarea *j* tiene un tiempo de procesado *pj ,* una fecha fin *dj* *,* un tiempo de completado *Cj ,* y un retraso de la tarea *Tj* . Hay que destacar que las propiedades *Cj y Tj*  son propiedades derivadas en cierta manera, *Cj*  es la suma de los tiempos procesados de todas las tareas que ya se han realizado más el tiempo de procesado de la tarea *j* así se consigue saber en cuanto tiempo se ha completado la tarea *j*, en el caso de *Tj*  este valor se consigue gracias a la siguiente fórmula *Tj* = max{0, *Cj - dj* }. El objetivo de este problema de optimización combinacional NP-duro es encontrar la secuencia de tareas que minimice el retraso total de todas las tareas, es decir, que la suma del retraso que existe entre todas las tareas se la menor posible, . Este problema tiene una variante en la que cada tarea tiene una importancia, es decir, un peso *wj* que afecta a la hora de calcular el retraso y es que la función objetivo a minimizar pasa a ser . Hay que destacar que este peso obviamente afectará a como se va a organizar la secuencia de tareas ya que determina una prioridad.

Partiendo de este problema de secuenciación vamos a usar el algoritmo ACO(Ant Colony Optimization) con distintas variantes para darle solución, y estudiar cómo se comporta el problema en cada algoritmo.