Universidad de los Andes

Departamento de Ingeniería civil y Ambiental.

"SCHEDULING: PROGRAMACIÓN LINEAL ENTERA IMPLEMENTANDO SOLVER DE EXCEL"

Proyecto de grado para aspirar al título profesional de:

Ingeniero Civil

Autor:

Natalia Prada Amaya

Asesor:

José Luis Ponz Tienda

Bogotá - Colombia

Julio de 2016

SCHEDULING: PROGRAMACIÓN LINEAL ENTERA IMPLEMENTANDO SOLVER DE EXCEL

TABLA DE CONTENIDO

INDICE DE TABLAS

INDICE DE ILUSTRACIONES

INDICE DE GRÁFICAS

RESUMEN

En el presente estudio se examinan dos metodologías específicas para la gerencia y programación de proyectos por medio de las formulaciones planteadas por dos principales autores, Pritsker y Ponz. Cada una de las teorías propuestas son analizadas para la implementación y programación lineal entera en Microsoft Excel, para el fácil desarrollo y ejecución por parte del lector.

Las dos teorías analizadas, RCPSP y RLP, describen estrategias para el manejo de recursos y duración de actividades de un proyecto de corta o gran magnitud, obteniendo resultados de minimización de tiempos finales, Makespan, y mejoras en la eficiencia de un proyecto gracias a la optimización de recursos, en donde se utilizan herramientas como Solver de Excel y Solver Premium instalado a Excel.

ABSTRACT

In the present research, two specific methodologies for project management and scheduling using the formulations posed by two main authors, Pritsker and Ponz are examined. Each of these theories proposed, are analyzed for implementation and integer linear programming (ILP) in Microsoft Excel, for easy development and performance by the reader.

The two theories analyzed, RCPSP and RLP, describe strategies for management of resources and the durations of project activities of short or large scale, obtaining minimization of expected completion time of the project (Makespan), and improvements in the efficiency thanks to the resource optimization using tools such as Excel Solver and Premium Solver.

PALABRAS CLAVES: Programación de proyectos; Microsoft Excel; Metodologías RCPSP y RLP; Duración de actividades; Optimización de recursos; Minimización y optimización de tiempos; Makespan; Eficiencia de un proyecto; Solver de Excel.

I. JUSTIFICACIÓN

En el ámbito de la construcción, unas de las necesidades más apremiantes al momento de la ejecución de una actividad, es determinar con gran exactitud la duración, los recursos a implementar y el momento de ejecución.

En la gerencia colombiana de proyectos de construcción, son pocas las empresas que cuentan con una programación adecuada que fomente el cumplimiento de cronogramas, la optimización de tiempos y recursos, por ello se han presentado retrasos, por ejemplo, en grandes obras de infraestructura colombiana (país.com.co, 2012). Con lo anterior, es posible promover la efectividad y calidad de los trabajos de una empresa, en donde el principal objetivo a futuro sea liderar el mercado.

Es muy común en proyectos en donde se implementa *Scheduling,* que se presenten aumentos y diferencias de grandes sumas de dinero entre el presupuesto inicial y el final. Esto se debe principalmente a una falla de estructuración y organización de tareas a ejecutar, en los cuales hay una serie de fallas: no se identifica la ruta de crítica de un proyecto, no se limita y/o optimiza la utilización de recursos por periodo, y en especial, no se estructuran los tiempos de holgura que le permitan a una actividad presentar un imprevisto sin afectar el cronograma y presupuesto global.

Por lo tanto, el principal interés de este proyecto se basa en la importancia de la implementación de *Scheduling* a un proyecto a partir de diferentes métodos, implementando las formulaciones de gran variedad de autores a partir de la herramienta tecnológica de *Solver* encontrada en *Microsoft Office Excel.*

II. INTRODUCCIÓN

En el presente trabajo se expondrán diferentes técnicas para implementar la teoría del *Scheduling* a través de estrategias aplicables a proyectos de amplia o corta magnitud, por medio de la implementación de programas como el Solver de Excel.

Existen diferentes metodologías para la programación de proyectos los cuales buscan optimizar algún aspecto en específico, ya sea el tiempo total de ejecución, manejo de costos de recursos, nivelación de estos, entre otros. Algunas de estas formulaciones son Enumeración por Ramificación, Acotación, TCTP[1], RCPSP[2] y RLP[3], y algunas más complejas como las Heurísticas constructivas y de mejora y Metaheurísticas.

Para la solución de problemas por medio de *Solver* de Excel se analizarán las metodologías del RCPSP (Resource – Constrained Project Scheduling Problems) y RLP (Resource Leveling Problems) a partir de las formulaciones propuestas por Pritsker en 1969 y Ponz Tienda en el 2013.

Así mismo, se desarrollará cada formulación aplicándolos a proyectos de diferentes magnitudes y cantidades de recursos y de actividades, evidenciando así su efectividad a través de un procedimiento detallado y explícito.

III. OBJETIVOS

III.1. Objetivo general

Por medio de este trabajo se pretende brindar y mostrar el funcionamiento de herramientas computacionales para dar resolución a problemas de optimización de tiempo y recursos en proyectos de construcción de gran envergadura. Las herramientas empleadas para cumplir con dicho objetivo son el Solver de Excel y Solver Premium para Excel.

[1] Por sus siglas en inglés Time-Cost Trade-Off Problem.
[2] Por sus siglas en inglés Resource Constrained Project Scheduling Problems.
[3] Por sus siglas en inglés Resource Leveling Problem.

III.2. Objetivos específicos

- Mostrar al lector la importancia de implementar herramientas computacionales para dar solución a problemas de la gerencia de proyectos, los cuales no podrían ser resueltos por medios convencionales.
- Desarrollar formulaciones lógicas para dar solución a cierto tipo de problema por medio de dos alternativas distintas, como lo son el RCPSP y RLP.
- Brindar herramientas al lector que le permitan determinar cuál de las alternativas desarrolladas a lo largo del trabajo es la ideal según las necesidades del proyecto y su gerente. Las necesidades pueden basarse en la eficacia del proyecto o la optimización de tiempos, lo cual, cada camino puede conllevar a optimizar costos.

IV. METODOLOGÍA

El desarrollo de este trabajo tuvo diferentes etapas como la de concepción y desarrollo. Durante la etapa de concepción, se planteó el alcance del trabajo al determinar cuáles eran los objetivos que se querían alcanzar para el mismo. Acto seguido, se determinó la forma en que podrían alcanzar los objetivos. Lo anterior con el fin de establecer los parámetros bajo los cuales se desarrollaría el trabajo durante su elaboración. De igual forma, se determinó el alcance del mismo al plantear las herramientas disponibles para alcanzar los objetivos.

Finalizada la etapa de concepción, se construyó una bibliografía basándose en los trabajos de autores que habían trabajado previamente las metodologías a estudiar, con el fin de implementar este material recogido computacionalmente, de tal forma que fuera posible encontrar soluciones a problemas de determinadas características por medios computacionales, lo cual haría el proceso de solución mucho más eficiente y óptimo que los métodos tradicionales.

Con el fin de ilustrar al lector formas de solucionar problemas de la gerencia de proyectos por medio de herramientas computacionales, se tomó el conocimiento adquirido de los autores estudiados y fue adaptado para desarrollar una formulación en el programa Microsoft Excel. Por medio de las formulaciones lógicas construidas para que su funcionamiento fuera computacional, se realizó la tarea de solucionar problemas planteados previamente por los autores, evidenciando la eficacia de la implementación de herramientas computacionales para dar solución a este tipo de problemas.

Con esto se pretendía demostrar las ventajas que representa la implementación de herramientas como Excel y Solver frente a metodologías convencionales para solucionar problemas de la gerencia de proyectos.

V. MARCO TEÓRICO

V.1. Project Scheduling

La programación de proyectos o Project Scheduling como es comúnmente conocido, consiste básicamente en la planificación, programación y control de las actividades presentes en un proyecto para lograr objetivos de rendimiento de costos y tiempo bajo la premisa de un uso de recursos de forma eficaz y eficiente.

En la planificación y programación de proyectos, primero se hacen principalmente las estimaciones de las necesidades brutas de los diferentes tipos de recursos, así como el cálculo aproximado de las duraciones y costos de la variedad de actividades que componen a un proyecto. Segundo, en la programación se busca la organización de las actividades concluyentes a desarrollar en el orden de tiempo en cual es necesario llevarse a cabo.

A partir de esto, los recursos necesarios y calculados a implementar por actividad, involucran una fase principal de la programación de proyectos ya que determinan el rendimiento real de un proyecto una vez ha iniciado su ejecución, por lo cual, son el factor determinante para la optimización y el buen funcionamiento del mismo.

Por lo tanto, el éxito principal de un proyecto y su ejecución formidable, radica en su finalización a tiempo y en el cumplimiento de presupuestos y especificaciones preestablecidas, sin embargo en los diferentes tipos de proyectos es muy común violar cualquiera de estos tres principales aspectos (Demeulemeester, 2002), y este el motivo principal por el cual es vital el *Project Scheduling* para la ejecución formidable de un proyecto.

1. CAPÍTULO I: RESOURCE – CONSTRAINED PROJECT SCHEDULING PROBLEMS (RCPSP)

En el entorno mundial, existe toda clase de proyectos que manifiestan problemáticas que afectan a gran escala su desarrollo y su efectivo funcionamiento. Comúnmente, los proyectos, como en el área de la construcción, se están enfocando cada día más en la optimización del tiempo y los recursos, debido a su importancia en la generación de beneficios al menor riesgo.

Este riesgo, se puede interpretar como aquello que se desea perfeccionar, minimizar, maximizar, entre otros, y que afecta a gran escala a un proyecto según su capacidad, la cual puede ser interpretada como el recurso principal a utilizar, dinero, dotaciones, equipo, personal etc.

De esta forma, se han estudiado a lo largo de los años, por una gran variedad de autores, metodologías que a partir del análisis de casos adoptados a diferentes áreas de trabajo, como la construcción, la educación, la salud, entre otras ramas, utilizando variedad de tareas y recursos para cada caso, se han creado diferentes teorías y formulaciones matemáticas para los problemas de programación de proyectos *(Project Scheduling)*.Por lo tanto, al ser una correcta planeación, programación y manejo de un proyecto tan importante como el proyecto mismo, a lo largo de este capítulo se propone un modelo práctico de modelación del RCPSP, bajo una simple implementación de solver de Microsoft Excel.

Tabla 1.1. Formulación matemática propuesta por (Pritsker, 1969).

Aspectos para la formulación matemática de RCPSP (Pritsker, 1969)

Objetivo deseado	*Restricciones*
1. Minimizar el tiempo total de ejecución de un proyecto	**1.** Disponibilidad limitada de Recursos **2.** Respetar las relaciones de precedencia entre las tareas **3.** No fragmentar la ejecución de las tareas

1.1. Contexto, antecedentes y formulaciones para el RCPSP

La primera formulación conocida para resolver la programación de proyectos por medio del RCPSP de forma óptima, fue desarrollada en la década de 1970 por Pritsker. Este autor propuso un modelo basado en una variable binaria x_{it} que establece el periodo de terminación de las actividades que componen a un proyecto cuando esta toma el valor de 1.

A partir del modelo de Pritsker, se han propuesto diferentes alternativas que adoptan e introducen nuevos términos a la metodología binaria. A finales de la década de 1980, Kaplan redefine los parámetros de x_{it} y propone una nueva descripción basada en el desarrollo y progreso de una tarea, en donde se define los parámetros de 1 y 0 cuando las actividades se están procesando y para cuando no, respectivamente. Sin embargo, este método puede presentar complicaciones para las tareas de inicio y fin, debido a que no poseen duración alguna, por lo tanto, en estos casos se puede adoptar una duración arbitraria de 1 y en las ocasiones donde se presente una actividad con duración cero, esta simplemente se elimina.

Para finales del siglo XX, Álvarez-Valdés y Tamarit (1993) y (Mingozzi A, 1998) proponen sus propias modificaciones de la programación de Pritsker.

Álvarez-Valdés y Tamarit transforman a la variable x para que su función este ligada a las relaciones de precedencia entre actividades y no al desarrollo o terminación de la misma. Esta variable se denomina x_{ij} y su decisión binaria 1 se establece cuando una actividad i es predecesora de una actividad j, de lo contrario el valor será igual a 0.

Así mismo, Mingozzi propuso su propia programación lineal utilizando subconjuntos a partir de dos variables binarias x_{it} y y_{jt}. Estas variables manejan los parámetros propuestos por Pritsker y Kaplan, en donde x_{it} asume valores de 1 cuando el posible subconjunto i esta siendo procesado en un periodo t y y_{jt} esta determinado por la metodología inicial de Pritsker, en donde se asumen valores de 1 cuando una actividad j ha finalizado en un periodo t. Este método se justifica bajo la noción de posibles subconjuntos de actividades especificando relaciones de precedencia que, dependiendo el caso de programarse en paralelo, se deben respetar las restricciones de recurso que existan.

Para finalizar, en el año 2000 una formulación muy similar a la de Kaplan es propuesta por (Klein, 2000). Esta formulación plantea una nueva variación a la variable x_{it} en donde la condición binaria se establece no sólo cuando se está desarrollando o procesando en un periodo t sino también será igual a 1 si en un instante anterior lo estuvo. Por lo tanto, existirá una serie de ceros seguida por una secuencia de 1.

1.1.1. Formulación de Pritsker

La programación de proyectos propuestos por Pritsker en 1969, determina principalmente el momento en el cual una actividad debe dar inicio y debe ser procesada con una disponibilidad de recursos limitada. Esta formulación se basa en la implementación de un sistema binario (0-1) que indica en un determinado periodo, si la actividad que se desea desarrollar ha culminado o no.

El principal reto a superar consta en realizar una eficiente formulación y definición de las variables, de las restricciones y de una correcta función objetivo. Por esto mismo, encontrar los periodos de tiempo en los cuales una actividad deba procesarse, depende exclusivamente de los objetivos deseados para el proyecto.

1.1.2. Modelo de Pritsker

Para ilustrar el periodo de desarrollo de una tarea específica de un proyecto se propone la implementación de la variable binaria x_{it} de tal forma que si una actividad i finaliza en el tiempo t adopta el valor de 1 y en caso contrario cero [1.1].

$$x_{it} \in \{0,1\} | i \in \{\, i = 1,2, \dots N\}, t \in \{1,2, \dots T\}$$

siendo

$$N = \{i, i+1, \dots N\} = número\ de\ actividades \qquad [1.1]$$

$$T = \{t, t+1, \dots T\} = cota\ conocida\ del\ problema$$

Dado que las actividades de un proyecto tan solo pueden tener un periodo de finalización, se impone que la sumatoria de todas las variables x_{it} de una actividad deberá ser igual a 1 [1.2].

$$\sum_{t=1}^{T} x_{it} = 1 \ \forall i \in N \qquad [1.2]$$

Con el objetivo de simplificar el modelo y reducir el número de variables, la ecuación [1.2] puede limitarse entre los tiempos más pronto y más tarde de terminar de cada una de las actividades. Por tal motivo, la ecuación [1.2] se expresa de la siguiente forma:

$$\sum_{t=EF_i+1}^{\square F_i+T-mk} x_{it} = 1 \qquad [1.3]$$

Donde EF_i es el tiempo más pronto de terminar de i, LF_i el tiempo más tarde de terminar de i, T es una cota conocida del problema y mk,o makespan del problema, es irrestricto, es decir, sin restricción de recursos, y calculado aplicando el tradicional *Forward and Backward pass* (Demeulemeester,2002).

Una vez determinados los valores de x_{it} y de t, existe un vínculo entre estas dos variables que permite encontrar el periodo exacto de terminación de una actividad i. Este tiempo de finalización puede ser calculado para cada actividad a partir de [1.4].

$$\sum_{t=EF_i+1}^{LF_i+T-mk} t \cdot x_{it} \qquad [1.4]$$

Bajo estas circunstancias, una actividad puede culminar tan tarde como la holgura, la duración y sus restricciones lo permitan.

Función Objetivo: Las diferentes tareas que componen a un proyecto son organizadas y programadas para optimizar su rendimiento según su limitación de recursos. El desarrollo de cada una de ellas está sujeta a ciertos requerimientos y limitaciones, las cuales determinan y encuentran una solución a la función objetivo. Para este caso de análisis, se analizará las diferentes formulaciones y restricciones posibles para la minimización del tiempo total de desarrollo de un proyecto.

Minimizar el tiempo total, que en pocas palabras significa minimizar el periodo en el cual todas las tareas de un proyecto se han completado, es un trabajo que está sujeto a ciertas restricciones.

1. **Finalización de una tarea:** Cada tarea posee un periodo exacto de finalización dado por la ecuación [1.4].

2. **Secuenciación:** Se presentan restricciones de secuenciación cuando una determinada tarea no se debe iniciar hasta que su predecesora o predecesoras se hayan desarrollado en su totalidad. Si una actividad n debe preceder a una actividad m, lo periodos t para cada una estarían sujetos a

$$t_m \cdot x_{mt} \leq t_n \cdot x_{nt} - d_n \qquad [1.5]$$

en donde, cada tiempo de t_m y t_n están dados por la ecuación [1.4],

$$\sum_{t=EF_i+1}^{LF_i+T-mk} t_m \cdot x_{mt} \leq \sum_{t=EF_i+1}^{LF_i+T-mk} t_n \cdot x_{nt} - d_n \qquad [1.6]$$

3. **Restricciones de Recursos:** Para el desarrollo de los diferentes trabajos que componen a un proyecto, en muchos casos, se manejan restricciones de recursos en donde existe una cantidad limitada R que supone el número máximo por periodo permitido para el rendimiento en un proyecto.

Este valor de restricción R, no puede llegar a ser excedido por la cantidad de recursos utilizados en cada periodo de tiempo. De esta forma las restricciones de recursos para un proyecto esta formulado como:

$$\sum_{i=1}^{N} r_{it} \leq R \qquad [1.7]$$

4. **Función Objetivo:** Para finalizar, la función objetivo se basa principalmente en el principio de minimizar el makespan del proyecto.

$$Mínimo \sum_{t=EF_i+1}^{LF_i+T-mk} t \cdot x_{it} \quad , i = N \qquad [1.8]$$

Quedando la formulación completa de la siguiente forma:

$x_{it} \in \{0,1\} \mid i \in \{ i = 1,2, \dots N\}, t \in \{1,2, \dots T\}$ $N = \{i, i+1, \dots N\} = número\ de\ actividades$ $T = \{t, t+1, \dots T\} = cota\ conocida\ del\ problema$	[1.1]
$\sum_{t=1}^{T} x_{it} = 1\ \forall i \in \{1,2, \dots T\}$	[1.2]
$\sum_{t=EF_i+1}^{LF_i+T-mk} x_{it} = 1$	[1.3]
$\sum_{t=EF_i+1}^{LF_i+T-mk} t \cdot x_{it}$	[1.4]
$t_m \cdot x_{mt} \leq t_{n \cdot x_{nt}} - d_n$	[1.5]

$$\sum_{t=EF_i+1}^{LF_i+T-mk} t_m \cdot x_{mt} \leq \sum_{t=EF_i+1}^{LF_i+T-mk} t_n \cdot x_{nt} - d_n \qquad [1.6]$$

$$\sum_{i=1}^{N} r_{it} \leq R \qquad [1.7]$$

$$Mínimo \sum_{t=EF_i+1}^{LF_i+T-mk} t \cdot x_{it} \quad , i = N \qquad [1.8]$$

1.1.3. Implementación usando Solver de Microsoft Excel y Solver Premium

Para el desarrollo del método utilizando las diferentes herramientas de Solver, es necesario analizar el proyecto considerando cada una de sus actividades (tiempo, duración, recursos y relaciones de precedencia definidas), encontrando así la programación adecuada para la mínima duración total del proyecto a partir de la asignación de los tiempos de inicio de cada actividad.

Para iniciar el desarrollo del método de Pritsker, se implementa la siguiente técnica a partir de un modelo aplicable y programable a cualquier tipo de proyecto que se quiera analizar y optimizar, utilizando herramientas como Excel.

EJEMPLO 1.1. RCPSP

A. Planteamiento del problema:

Se tiene un proyecto dado, compuesto de N = 7 actividades, dos recursos con capacidades máximas de $R_1 = 4$ y $R_2 = 2$ y una cota máxima de $T = 17$.

B. Solución

1. La Tabla 2, que se presenta a continuación, tiene como objeto principal la organización de forma básica de tareas o labores necesarias que se involucran en el proyecto, determinando desde el inicio la duración, las relaciones de precedencia y la cantidad necesaria de recursos por actividad.

Tabla 1.2. Determinación de actividades, duraciones, recursos y relaciones de precedencia

ACTIVIDAD	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
DURACIÓN	0	3	4	6	2	1	4	0
R1	0	2	2	2	2	3	3	0
R2	0	1	1	1	1	0	1	0
PREDECESOR INMEDIATO	-	0	0	1,2	2	3,4	4	5,6

2. Considerando la Tabla 1 y teniendo todos los parámetros establecidos, se recomienda examinar el problema bajo la implementación de un gráfico de nodos, de tal forma que se facilite el análisis del proyecto.

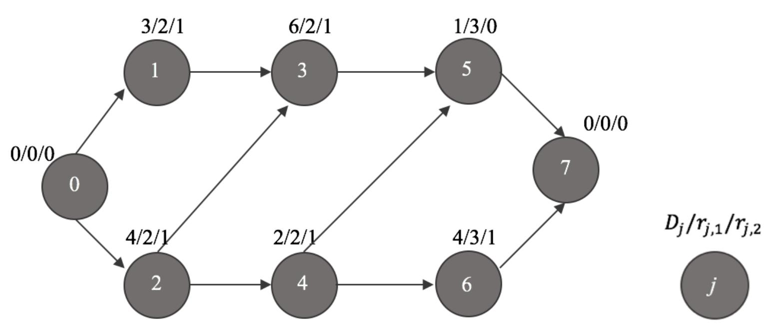


Gráfico 1.1 Representación de relaciones predecesoras en gráfico de nodos (Proon, 2010) ejemplo 1.1.

3. Usando los valores de duración para todas las actividades, se calcula los tiempos más tardíos de empezar y de terminar, ES_i (Early Start), EF_i (Early Finish), LS_i (Late Start), LF_i (Late Finish), y el makespan del proyecto (con el método de *Forward and Backward pass)* utilizando las relaciones de precedencia. Se debe tener en cuenta que bajo el planteamiento de Pritsker solo se utilizan los valores de los tiempos más pronto y tarde de terminar de cada actividad.

Tabla 1.3. Tabla de determinación de ES, EF, LF Y LF

ACTIVIDAD	DURACIÓN	ES	EF	LS	LF
0	0	0	0	0	0
1	3	0	3	1	4
2	4	0	4	0	4
3	6	4	10	4	10
4	2	4	6	5	7
5	1	10	11	10	11
6	4	6	10	7	11
7	0	11	11	11	11
MAKESPAN			**11**		

4. Si se sabe la cota máxima de duración (T) que se le puede permitir al proyecto, se calcula el período total de holgura para cada actividad como la diferencia del tiempo máximo y el makespan.

Tabla 1.4. Holgura máxima para el desarrollo del proyecto

Makespan	11
T	17
T - Makespan	**6**

5. Por medio de Microsoft Excel se desarrolla una tabla que contenga todos los datos anteriormente mencionados y calculados, efectuando los formatos condicionales necesarios.

Tabla 1.5. Tabla de holguras y tiempos permitidos en el que se puede desarrollar una actividad.

ACTIVIDAD	EF	LF	Σxij	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	3	4	1			1	0	0	0	0	0	0	0									
2	4	4	1				1	0	0	0	0	0	0									
3	10	10	1										1	0	0	0	0	0	0			
4	6	7	1						1	0	0	0	0	0	0	0						
5	11	11	1											1	0	0	0	0	0	0		
6	10	11	1										1	0	0	0	0	0	0	0		
7	11	11	1											1	0	0	0	0	0	0		
Makespan													11									

Momento en el que finaliza la actividad

Celdas con formato condicional 1.1.

FORMATO CONDICIONAL 1.1.

Para Microsoft Excel es necesario desarrollar dos reglas; 0 la primera determina las fronteras de la variable binaria x_{it} presentes en la ecuación [1.3], las cuales establece las casillas por donde se puede mover una actividad en el tiempo, y la segunda 1 debe encontrar todos los valores que sean iguales a 1.

ΣXit: Debe ser igual a la suma de todos los valores que se encuentren ubicados dentro de la holgura, ecuación [1.3].

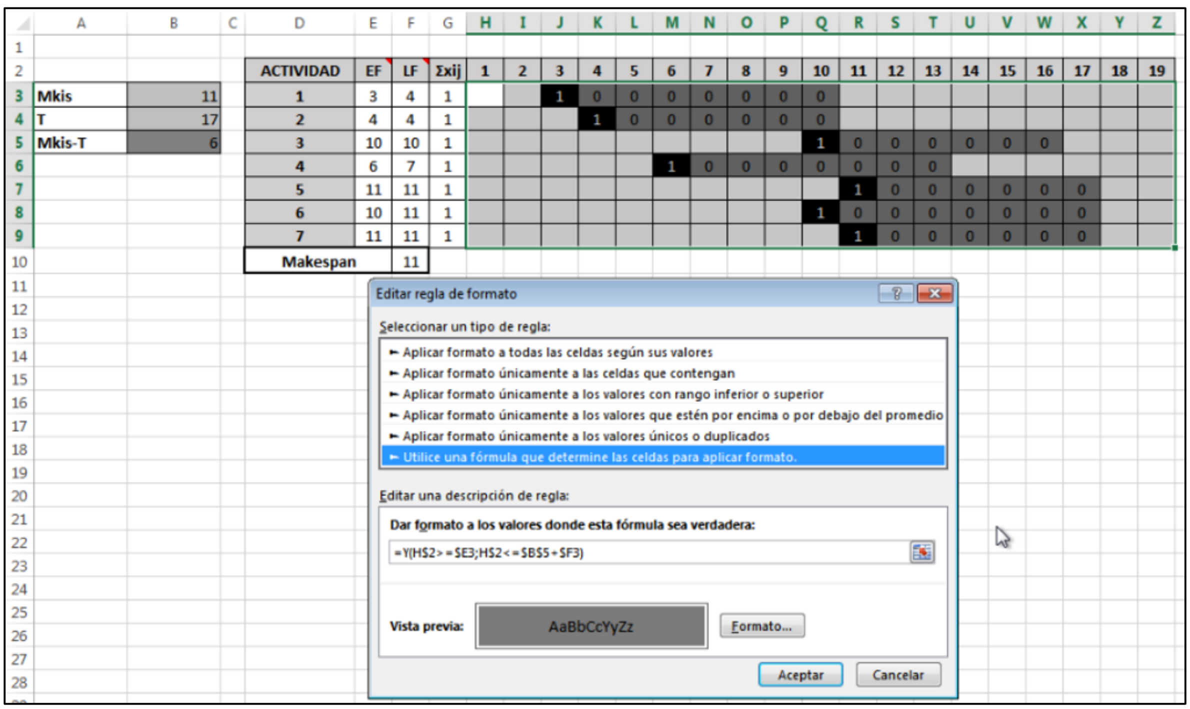


Ilustración 1.1. Formato condicional para holguras y tiempos permitidos

En la ilustración 1.1, se puede observar la implementación de la herramienta de formato condicional de Excel, la cual permite resaltar el rango de holgura que se tiene por actividad bajo la adaptación de la siguiente regla:

$$Y(t_i \geq EF_i; t \leq LF_i + T - Mk) \quad [1.9]$$

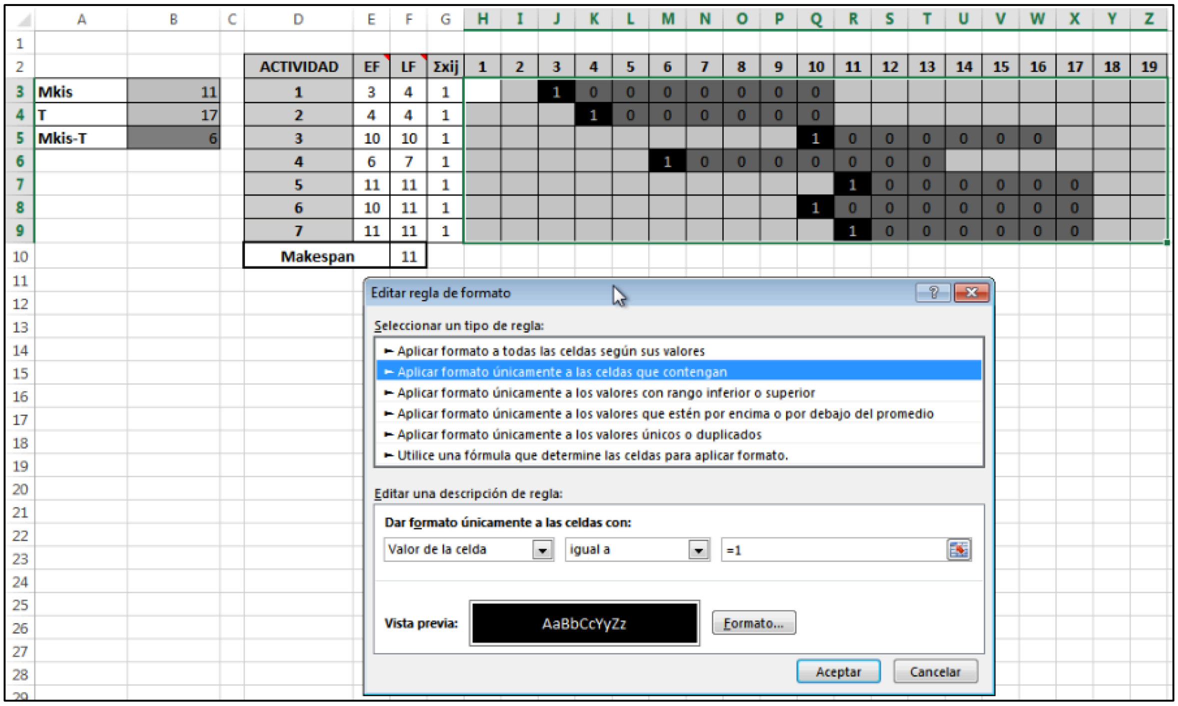


Ilustración 1.2. Formato condicional que encuentra el momento de finalización de cada actividad

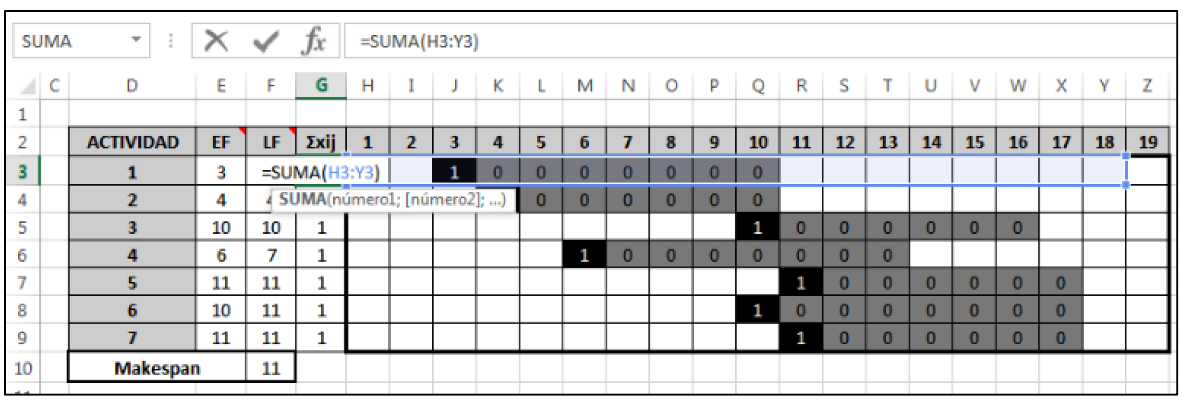


Ilustración 1.3. Formato condicional para la suma de periodos de una actividad (ΣXit)

Se puede observar en la ilustración 1.4. que las celdas resaltas son las variables binarias que darán posteriormente solución al ejercicio.

6. Se desarrolla por medio de una tabla, el gráfico de nodos de la ilustración 1, en donde se determinan los tiempos de fin e inicio de cada una de las actividades predecesoras y sucesoras respectivamente.

Tabla 1.6. Gráfico de Nodos (ilustración 1) representado de forma tabular.

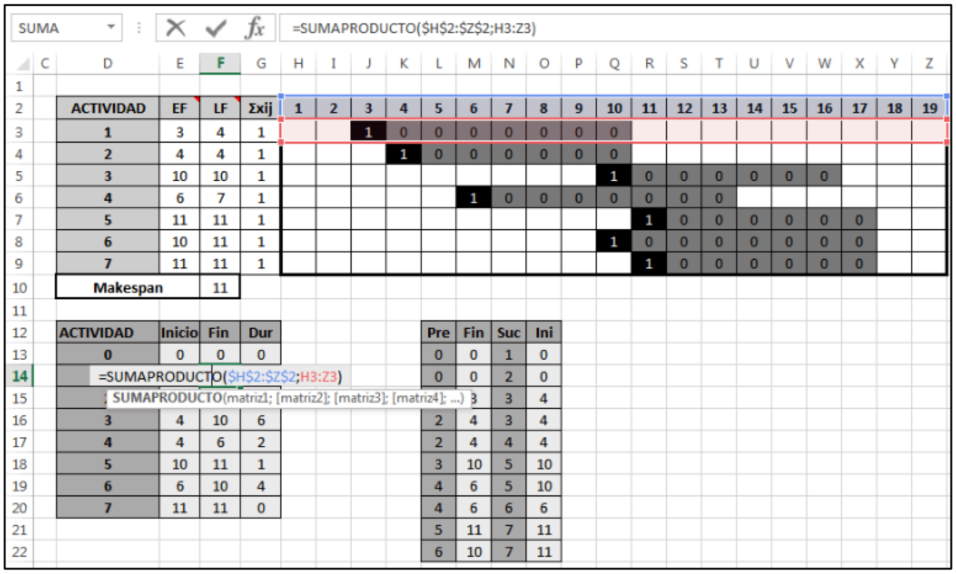
Predecesora	Fin	Sucesora	Inicio
0	0	1	0
0	0	2	0
1	3	3	4
2	4	3	4
2	4	4	4
3	10	5	10
4	6	5	10
4	6	6	6
5	11	7	11
6	10	7	11

Cada tiempo de inicio y fin de la tabla 1.6. debe estar relacionado directamente con el momento en el que finaliza cada actividad (ver tabla 1.5). Para ello, se implementa la función de BUSCARV en cada tarea a partir de la implementación de una nueva tabla (ver tabla 1.7) que contenga los periodos de desarrollo por actividad, (ilustración 1.7 y 1.8).

Tabla 1.7. Tiempos de inicio y fin para cada actividad

ACTIVIDAD	Inicio	Fin	Dur
0	0	0	0
1	0	3	3
2	0	4	4
3	4	10	6
4	4	6	2
5	10	11	1
6	6	10	4
7	11	11	0

La tabla 1.7. debe estar estrechamente ligada a x_{it}, en donde, la aplicación de la herramienta SUMAPRODUCTO, nos permite encontrar el periodo de finalización de cada actividad de igual forma que lo planteaba Pristker, (ver ecuación [1.2].)



=SUMAPRODUCTO($H$2:$Z$2;H3:Z3)

ACTIVIDAD	EF	LF	Σxij	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	3	4	1			1	0	0	0	0	0	0	0									
2	4	4	1				1	0	0	0	0	0	0									
3	10	10	1										1	0	0	0	0	0	0			
4	6	7	1						1	0	0	0	0	0	0	0						
5	11	11	1											1	0	0	0	0	0	0		
6	10	11	1										1	0	0	0	0	0	0	0		
7	11	11	1											1	0	0	0	0	0	0		
Makespan		11																				

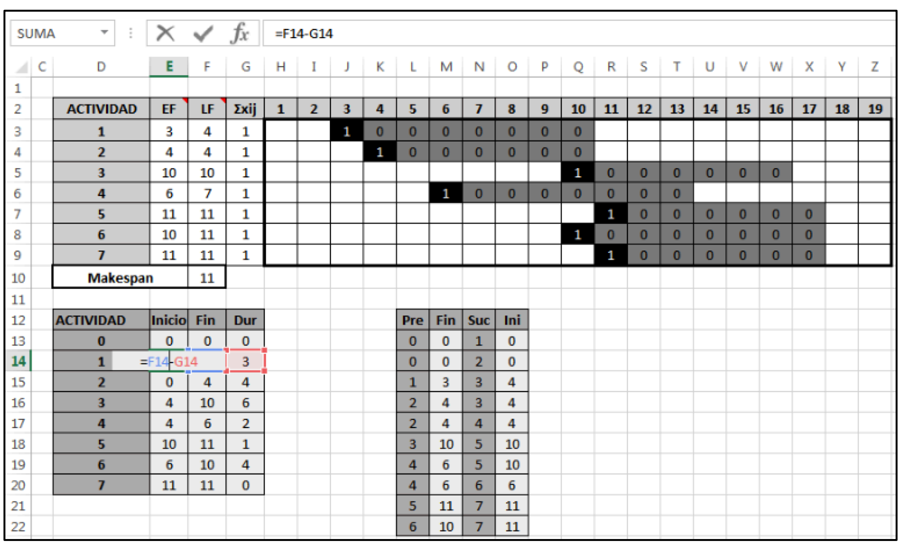
ACTIVIDAD	Inicio	Fin	Dur
0	0	0	0
=SUMAPRODUCTO($H$2:$Z$2;H3:Z3)			
SUMAPRODUCTO(matriz1; [matriz2]; [matriz3]; [matriz4]; ...)			
3	4	10	6
4	4	6	2
5	10	11	1
6	6	10	4
7	11	11	0

Pre	Fin	Suc	Ini
0	0	1	0
0	0	2	0
	3	3	4
2	4	3	4
2	4	4	4
3	10	5	10
4	6	5	10
4	6	6	6
5	11	7	11
6	10	7	11

Ilustración 1.4. Parámetros iniciales de finalización para cada actividad

Los valores de inicio de la tabla 1.7 se calculan fácilmente como la sustracción entre el fin de una actividad y su duración.

$$Inicio_i = Fin_i - duración_i$$



=F14-G14

ACTIVIDAD	EF	LF	Σxij	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	3	4	1			1	0	0	0	0	0	0	0									
2	4	4	1				1	0	0	0	0	0	0									
3	10	10	1										1	0	0	0	0	0	0			
4	6	7	1						1	0	0	0	0	0	0	0						
5	11	11	1											1	0	0	0	0	0	0		
6	10	11	1										1	0	0	0	0	0	0	0		
7	11	11	1											1	0	0	0	0	0	0		
Makespan		11																				

ACTIVIDAD	Inicio	Fin	Dur
0	0	0	0
1	=F14-G14		3
2	0	4	4
3	4	10	6
4	4	6	2
5	10	11	1
6	6	10	4
7	11	11	0

Pre	Fin	Suc	Ini
0	0	1	0
0	0	2	0
1	3	3	4
2	4	3	4
2	4	4	4
3	10	5	10
4	6	5	10
4	6	6	6
5	11	7	11
6	10	7	11

Ilustración 1.5. Calculo de tiempos de inicio por actividad

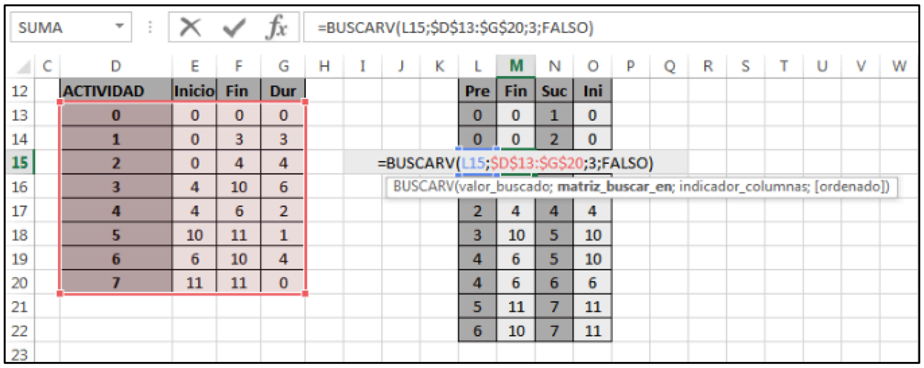


Ilustración 1.6. Determinación de valores de precedencia

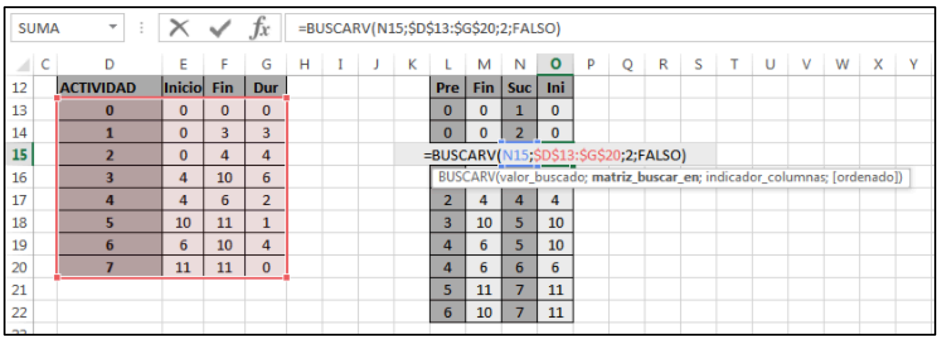


Ilustración 1.7. Determinación de valores de precedencia

7. Por último, utilizando el formato condicional 1.2., se crea el calendario de recursos por cada recurso que se tenga como restricción, en donde debe ser posible identificar la cantidad total empleada por periodo.

Tabla 1.8. Calendario de recursos R1.

	Cantidad Recurso R_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
4	2	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
5	3	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
6	3	0	0	0	0	0	0	3	3	3	3	0	0	0	0	0	0	0	0	0
7	0																			
TOTAL RECURSOS		4	4	4	2	4	4	5	5	5	5	3	0	0	0	0	0	0	0	0
DISP	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

Formato condicional 1.2.

FORMATO CONDICIONAL 1.2.

Se debe desarrollar una regla que determine la cantidad de recursos a utilizar de una actividad a lo largo de toda su duración.

Tabla 1.9. Calendario de recursos R2.

	Cantidad Recurso R_2	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL RECURSOS		2	2	2	1	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
DISP	**2**	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

Este formato condicional aplicado a cada una de las celdas que componen el calendario de recursos maneja ciertas condiciones. Para dichas condiciones, durante el desarrollo de cada tarea del proyecto se debe observar la cantidad de recursos que se está utilizando en cada periodo.

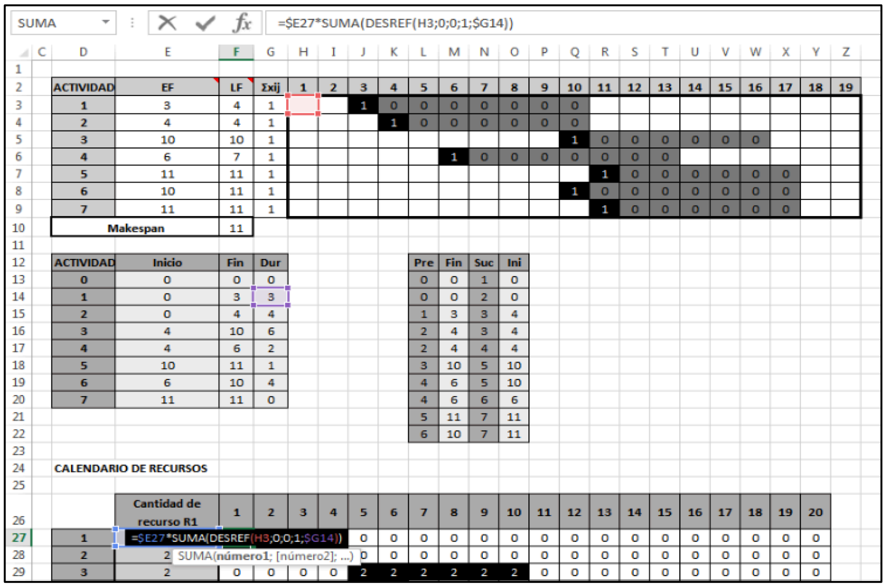


Ilustración 1.8. Formulación calendarios de recursos

C. Restricciones

Para la implementación del método de RCPSP de Pritsker por medio de la herramienta Solver de Microsoft Excel y a partir del desarrollo tabular previamente expuesto, es necesario establecer las siguientes restricciones:

1. Se desean modificar o variar las celdas que son positivas o acertadas en el formato condicional (ver tabla 2).

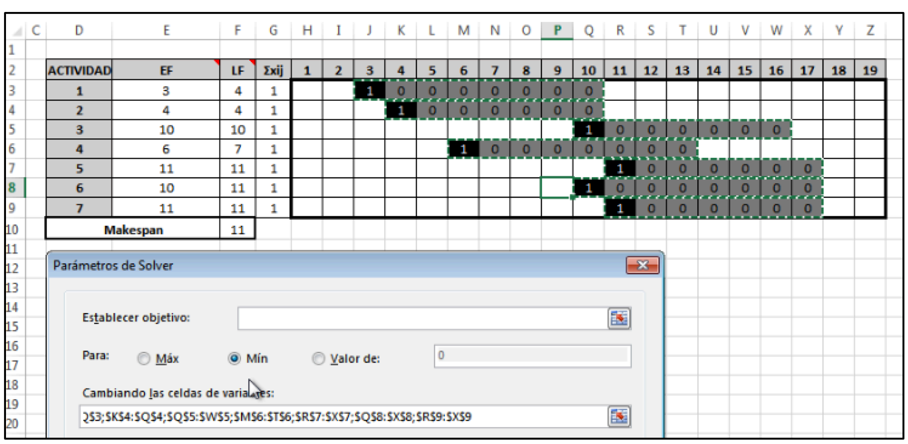


Ilustración 1.9. Celdas variables por actividad

2. Las celdas que se encuentren positivas dentro del formato condicional 1.1.(ver tabla 2) deben ser binarias de acuerdo a la ecuación [1.1].

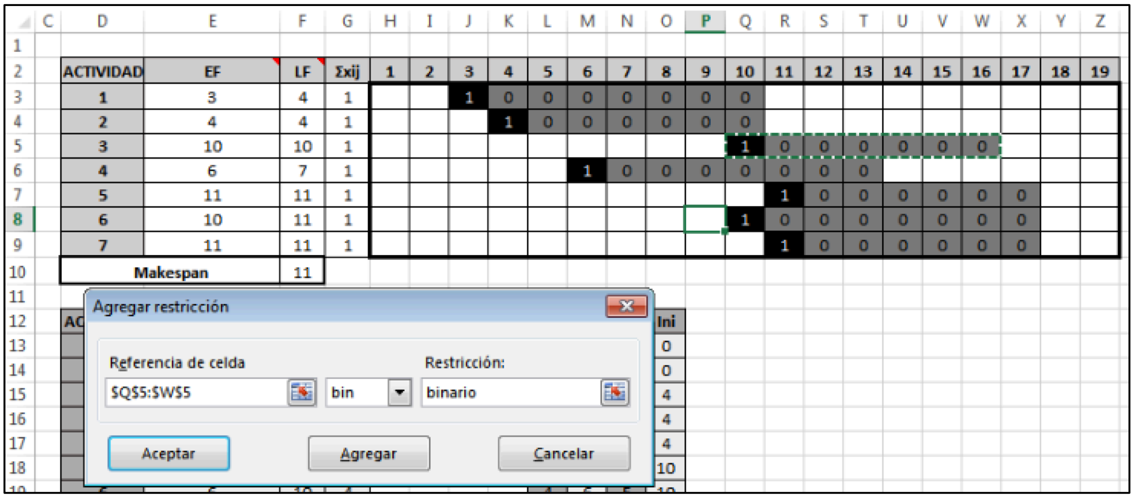


Ilustración 1.10. Determinación de celdas binarias por actividad.

3. La sumatoria *ΣXit (*ver tabla 2) siempre debe ser igual a 1 de acuerdo a la ecuación [1.3] .

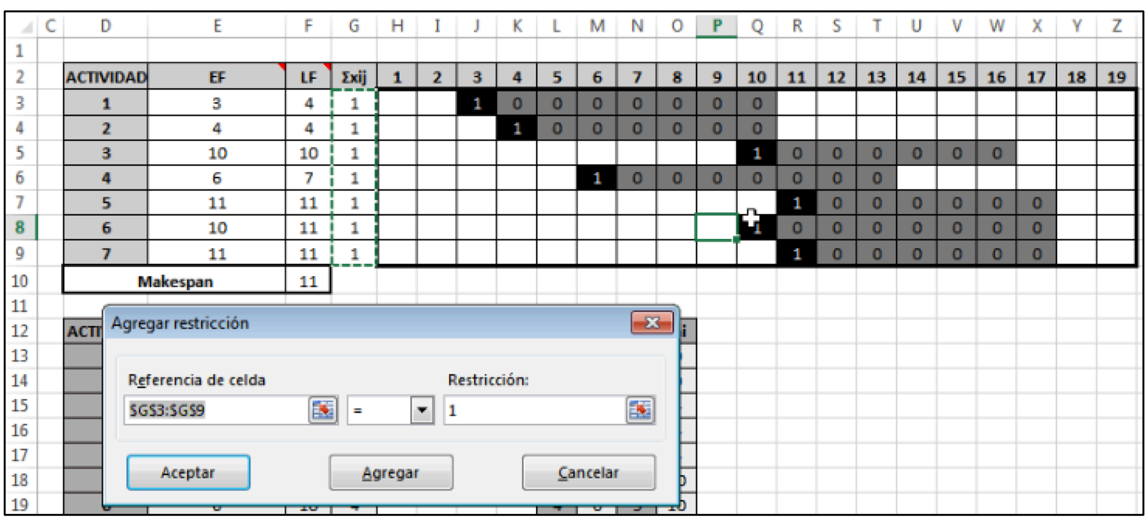


Ilustración 1.11. Selección de sumatoria de periodos por actividad asignando restricción de ΣXit=1

4. El inicio de una actividad sucesora debe ser mayor a la finalización de su actividad predecesora (ver tabla 3) de acuerdo a la ecuación [1.5] .

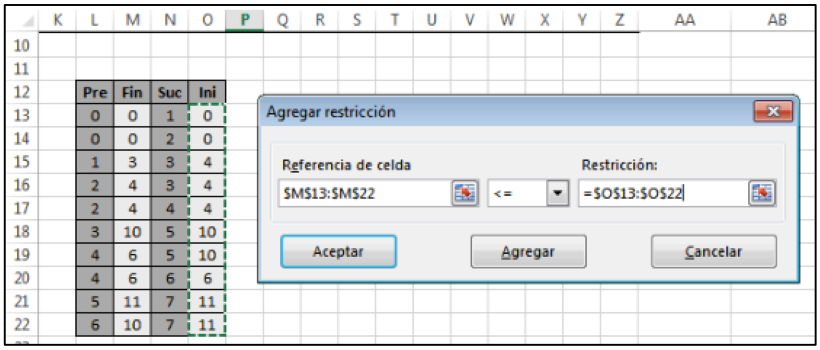


Ilustración 1.12. Determinación de restricciones de precedencia.

5. El total de recursos por periodo no debe ser mayor a la disponibilidad que se tiene para el proyecto (tabla 4), ecuación [1.7].

Ilustración 1.13. Restricción de recursos R_1.

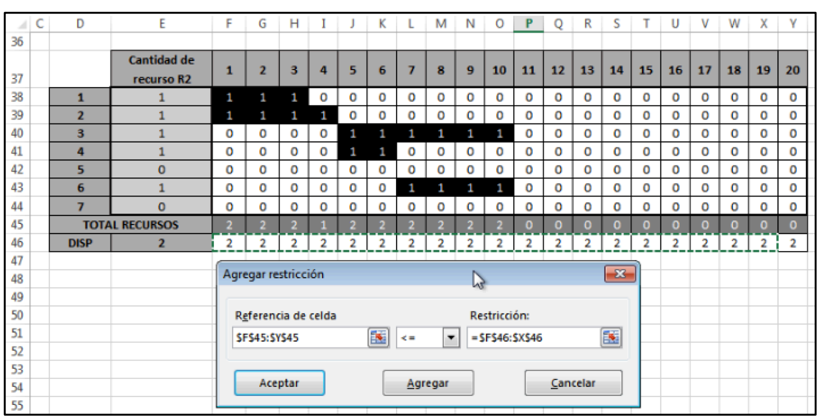


Ilustración 1.14. Restricción de recursos R_2

6. Por último, se desea minimizar la celda objetivo z, la cual representa el tiempo total de ejecución del proyecto, el cual se puede calcular como el tiempo de finalización de la última actividad del proyecto.

$$z = \sum_{t=EF_i+1}^{LF_i+T-mk} t \cdot x_{it} \quad , i = N$$

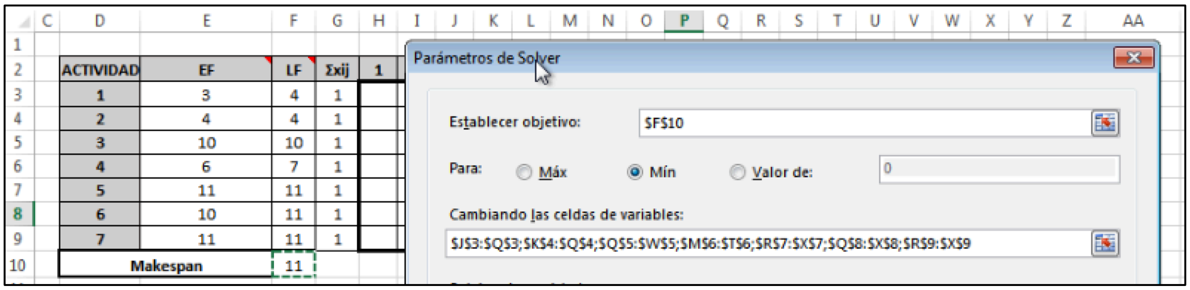


Ilustración 1.15. Parámetros de Solver, función objetivo.

Para finalizar, se ejecuta el programa utilizando el método de resolución propuesto por Solver, Simplex LP.

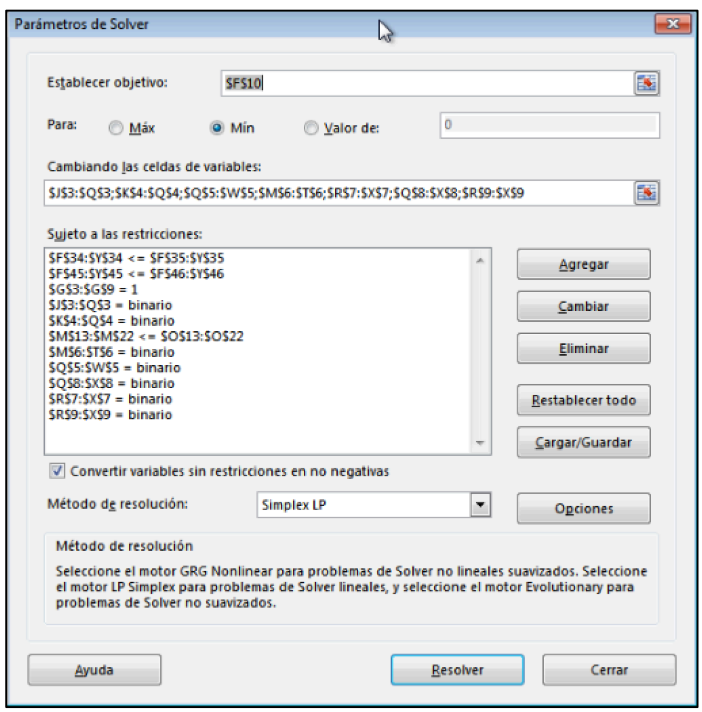


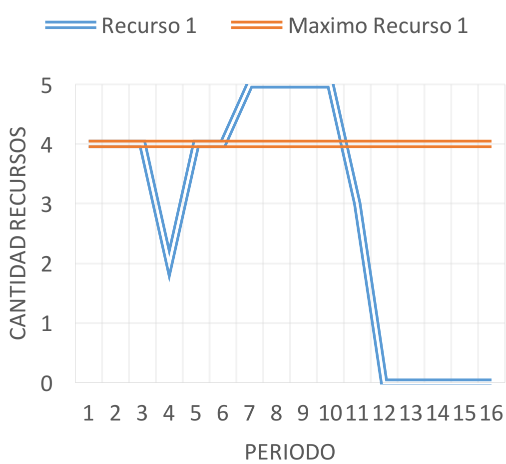
Ilustración 1.16. Parámetros de Solver con método de resolución Simplex LP.

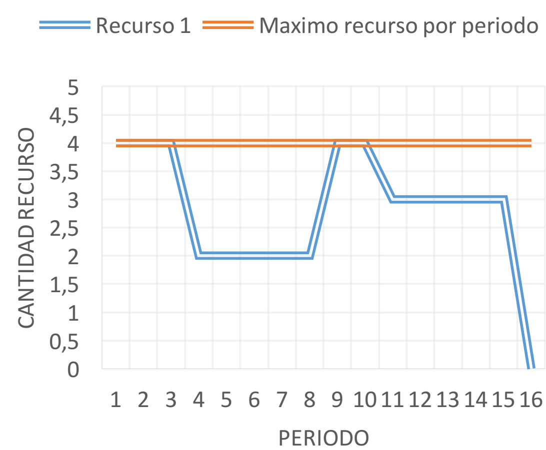
D. Resultados

Como resultado de la implementación del Solver y el cumplimiento de las restricciones, el programa nos arroja una nueva organización para el desarrollo de las actividades.

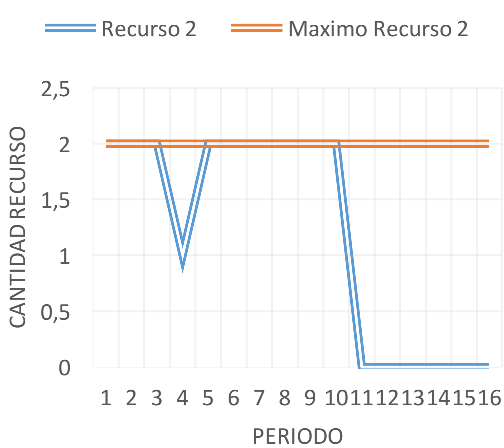
En las gráficas 1 y 2, se puede observar cómo se distribuyen nuevamente los recursos y se presenta una nivelación de cada uno de estos a pesar de que el makespan del proyecto aumente.

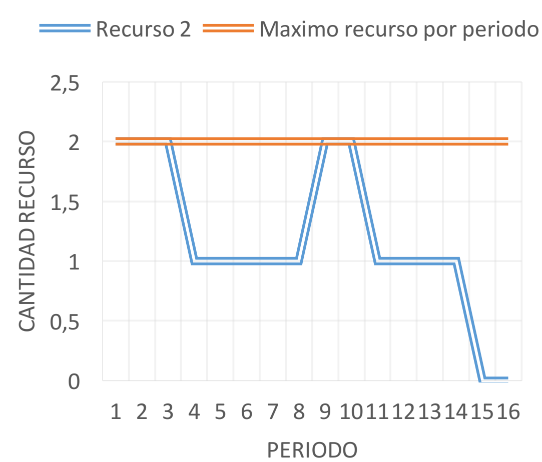
RECURSO 1	
Recurso 1 sin optimizar	Recurso 1 optimizado





RECURSO 2	
Recurso 2 sin optimizar	Recurso 2 optimizado





Los cambios presentes en cada uno de los gráficos demuestran el funcionamiento de la teoría y la formulación propuesta por Pritsker en el año 1969. De acuerdo a Pritsker, para aplicaciones como en los casos en donde si se tiene un plazo máximo para el proyecto, es posible nivelar los recursos a lo largo de este, de tal forma que se cumplan las restricciones de disponibilidad de recursos.

La implementación de herramientas como Excel facilita ampliamente la solución de problemas, no solo optimizando el desarrollo del proyecto, sino también la formulación del mismo.

A continuación, se presenta la nueva organización optimizada de las tareas en donde se cumplen cada una de las restricciones.

Tabla 1.10. Resultado final organización de actividades optimizado.

ACTIVIDAD	EF	LF	Σxij	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	3	4	1			1	0	0	0	0	0	0	0									
2	4	4	1				1	0	0	0	0	0	0									
3	10	10	1										1	0	0	0	0	0	0			
4	6	7	1						0	0	0	0	1	0	0	0						
5	11	11	1											0	0	0	0	1	0	0		
6	10	11	1										0	0	0	0	1	0	0	0		
7	11	11	1											0	0	0	0	1	0	0		
Makespan	15																					

EJEMPLO 1.2 (IMPLEMENTACIÓN SOLVER PREMIUM)

A. Planteamiento del problema:

Se tiene un proyecto de construcción compuesto de N= 20 actividades, seis recursos a utilizar y un plazo máximo de T=49, y se suministra la siguiente información.

Tabla 1.11. Duración, relaciones de precedencia y recursos del problema

Actividad	Duración	Predecesor	R1	R2	R3	R4	R5	R6
Inicio	0	-	-	-	-	-	-	-
A	6	Inicio	5	2	2	2	7	4
B	3	Inicio	3	5	2	3	9	6
C	4	A	2	4	4	2	3	1
D	6	Inicio	5	4	3	5	5	4
E	7	A, B	3	5	2	3	8	0
F	5	C	4	1	4	9	2	5
G	2	D	4	1	4	3	9	8
H	2	A, B	5	5	4	0	9	1
I	2	G, H	3	2	4	3	4	2
J	6	F	1	5	4	6	7	3
K	1	C, E	3	3	2	4	5	1
L	2	E, G, H	3	2	2	8	3	4
M	4	I, K	2	2	2	2	4	8
N	2	F, L	1	4	4	3	4	1
O	3	L	5	5	4	6	2	3
P	5	J, M, N	3	2	3	4	7	8
Q	8	O	4	5	4	2	3	4
R	2	D, O	5	3	3	3	7	8
S	6	P, R	2	4	6	2	3	4
T	2	Q	1	6	2	7	5	2
Fin	0	S, T	-	-	-	-	-	-
Limitante de recursos			7	10	10	16	18	13

B. Solución

Para el desarrollo del problema es necesario implementar el mismo procedimiento de plantilla y formulación de Excel presentado en el ejercicio anterior.

1. Según los datos presentados previamente (ver tabla 1.11), se desarrolla el gráfico de nodos correspondiente, cumpliendo cada una de las restricciones de precedencia establecidas.

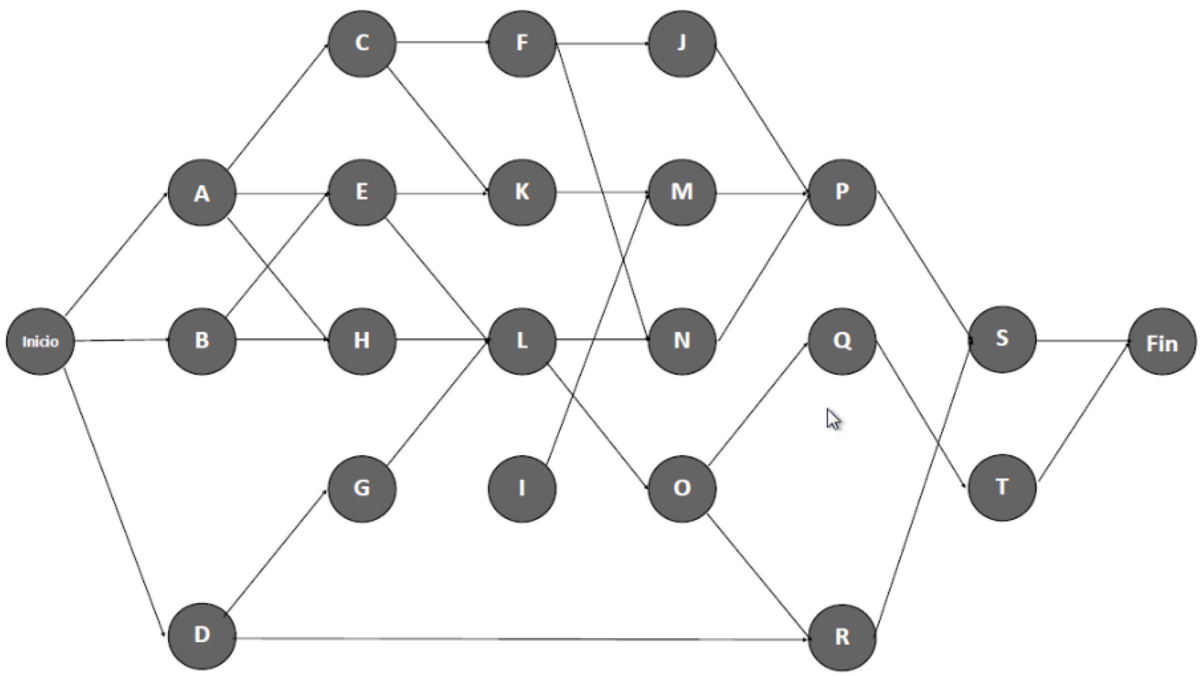


Gráfico 1.2. Gráfico de nodos del proyecto ejemplo 1.2.

2. Al igual que en el ejercicio anterior, se implementa el método de *Forward and Backward pass* con el fin de calcular cada uno de los valores necesarios en la implementación del método, los cuales son ES_i (Early Start), EF_i (Early Finish), LS_i (Late Start), LF_i (Late Finish), y el makespan del proyecto.

   Es importante resaltar que el makespan aquí calculado no es el óptimo.

Tabla 1.12. Calculo de ES, EF, LF Y LF

Actividad	Duración	ES	EF	LS	LF
A	6	0	6	0	6
B	3	0	3	6	9
C	4	6	10	6	10
D	6	0	6	7	13
E	7	6	13	9	16
F	5	10	15	10	15
G	2	6	8	13	15
H	2	6	8	13	15
I	2	8	10	15	17
J	6	15	21	15	21
K	1	13	14	16	17
L	2	13	15	17	19
M	4	14	18	17	21
N	2	15	17	19	21
O	3	15	18	19	22
P	5	21	26	21	26
Q	8	18	26	22	30
R	2	18	20	24	26
S	6	26	32	26	32
T	2	26	28	30	32
FIN	0	32	32	32	32
MAKESPAN	**32**				

3. Siguiendo el procedimiento presentado en el ejemplo 1.1, se calcula la cota máxima del problema a partir del Makespan y el plazo máximo de finalización por tarea.

Tabla 1.13. Calculo de la holgura por actividad del proyecto.

Makespan	32
T	49
Makes-T	17

4. Se desarrolla la tabla de formatos de condicionales y variables binarias con los valores de holgura de cada tarea a partir del formato condicional 1.1.

Tabla 1.14. Tabla de holguras y tiempos permitidos en el que se puede desarrollar una actividad.

	EF	LF	ΣXij	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	6	6	1						1																				
B	3	9	1			1																							
C	10	10	1										1																
D	6	13	1						1																				
E	13	16	1													1													
F	15	15	1															1											
G	8	15	1								1																		
H	8	15	1								1																		
I	10	17	1										1																
J	21	21	1																					1					
K	14	17	1														1												
L	15	19	1															1											
M	18	21	1																		1								
N	17	21	1																	1									
O	18	22	1																		1								
P	26	26	1																										1
Q	26	30	1																										1
R	20	26	1																				1						
S	32	32	1																										
T	28	32	1																										
FIN	32	32	1																										

	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
A																										
B																										
C																										
D																										
E																										
F																										
G																										
H																										
I																										
J																										
K																										
L																										
M																										
N																										
O																										
P																										
Q																										
R																										
S						1																				
T		1																								
FIN						1																				

5. Se representa el gráfico de nodos a partir de una tabla de tal forma que más adelante se puedan establecer las restricciones de precedencia.

Tabla 1.15. Tiempos de inicio y fin para cada actividad

	INI	FIN	DUR
A		6	6
B		3	3
C	6	10	4
D		6	6
E	6	13	7
F	10	15	5
G	6	8	2
H	6	8	2
I	8	10	2
J	15	21	6
K	13	14	1
L	13	15	2
M	14	18	4
N	15	17	2
O	15	18	3
P	21	26	5
Q	18	26	8
R	18	20	2
S	26	32	6
T	26	28	2
FIN	32	32	

Predecesora	Fin	Sucesora	Inicio
A	6	C	10
A	6	E	13
A	6	H	8
B	3	E	13
B	3	H	8
C	10	F	15
C	10	K	14
D	6	G	8
D	6	R	20
E	13	K	14
E	13	L	15
F	15	J	21
F	15	N	17
G	8	I	10
G	8	L	15
H	8	I	10
H	8	L	15
I	10	M	18
J	21	P	26
K	14	M	18
L	15	N	17
M	18	P	26
N	17	P	26
O	18	Q	26
O	18	R	20
P	26	S	32
Q	26	T	28
R	20	S	32
S	32	FIN	32
T	28	FIN	32

6. Se desarrollan los seis calendarios de recursos implementando nuevamente el formato condicional 1.2., el cual fue utilizado en el ejercicio anterior.

Tabla 1.16. Calendario de recursos R1.

	R1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	5	5	5	5	5	5	5																				
B	3	3	3	3																							
C	2							2	2	2	2																
D	5	5	5	5	5	5	5																				
E	3							3	3	3	3	3	3	3													
F	4											4	4	4	4	4											
G	4							4	4																		
H	5							5	5																		
I	3									3	3																
J	1																1	1	1	1	1	1					
K	3														3												
L	3														3	3											
M	2															2	2	2	2								
N	1																1	1									
O	5																5	5	5								
P	3																						3	3	3	3	3
Q	4																			4	4	4	4	4	4	4	4
R	5																			5	5						
S	2																										
T	1																										
SUMA		13	13	13	10	10	10	14	14	8	8	7	7	7	10	9	9	9	8	10	10	5	7	7	7	7	7
Disp		7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7

	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
A																										
B																										
C																										
D																										
E																										
F																										
G																										
H																										
I																										
J																										
K																										
L																										
M																										
N																										
O																										
P																										
Q																										
R																										
S	2	2	2	2	2	2																				
T	1	1																								
SUMA	3	3	2	2	2	2																				
Disp	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7

C. Restricciones (implementación solver Premium)

Debido a la magnitud y a la cantidad de variables a utilizar en el problema, no es posible encontrar una solución del ejercicio de una forma convencional implementando el procedimiento de Solver presentado anteriormente. Por tal motivo, es necesario emplear la herramienta de Solver Premium, la cual permite desarrollar ejercicios de mayor magnitud y se puede instalar y procesar con facilidad en Microsoft Excel.

Para desarrollar el ejercicio es necesario seguir el siguiente procedimiento ingresando a Solver Premium a través de la plataforma ANALYTIC SOLVER PLATAFORM:

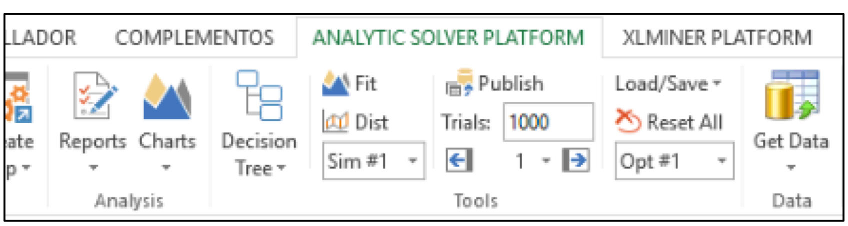


Ilustración 1.17. Plataforma de solver Premium ANALYTIC SOLVER PLATAFORM

1. Elegir las celdas variables del problema a partir de la opción *Variables* que se puede encontrar en el menú que se despliega en el costado derecho del programa, utilizando la función de *Add Normal Variable.*

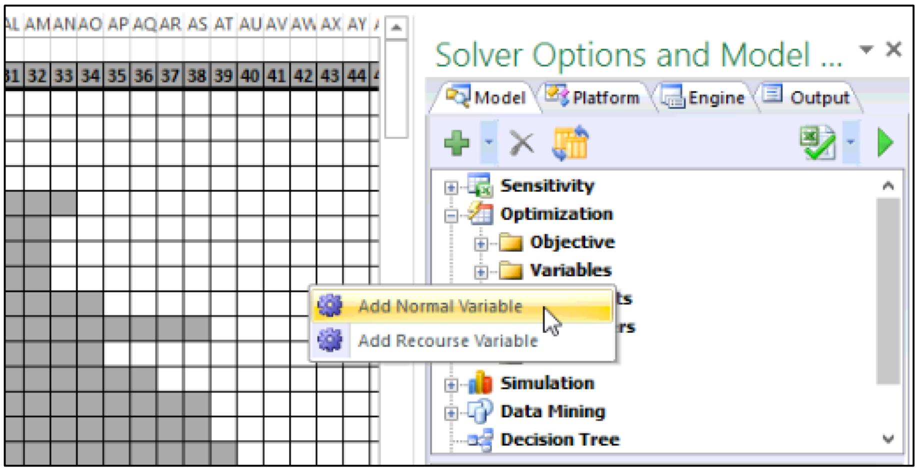


Ilustración 1.18. Ingreso de las variables binarias al programa.

2. Determinar cada una de las restricciones del problema utilizando la función *Constrains* ubicada en el menú superior. Esta función permite escoger el tipo de restricción que se está asignando en cada caso.

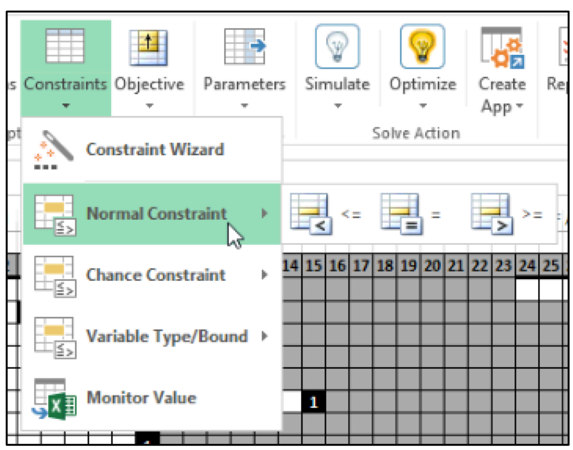


Ilustración 1.19. Ingreso de restricciones al programa.

A partir de esta función, se ingresan al programa las siguientes restricciones:

2.1. Sumatoria de celdas variables debe ser igual a 1, esto debido a su condición binaria.

2.2. El fin de las actividades predecesoras debe ser menor o igual al inicio de las actividades sucesoras.

2.3. Para cada calendario de recursos, se restringe la cantidad de recursos por periodo según las condiciones de disponibilidad que se presenten en el problema.

2.4. Las celdas variables del problema deben ser estrictamente binarias.

3. Una vez modelado el ejercicio, en el menú de *Solver Options and Model*, el cual se encontrará desplegado en el costado derecho del programa, se puede observar todos los parámetros ingresados.

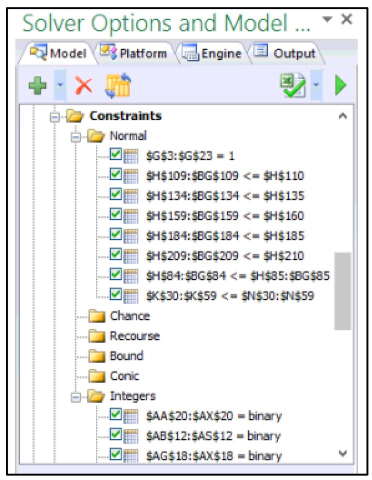


Ilustración 1.20. Menú de parámetros ingresados.

4. Por último, se escoge la función objetivo del problema a partir de la opción *Objective* ubicada en el menú superior. Para este caso la función objetivo es una variable normal irrestricta la cual se desea minimizar a partir de la ecuación [1.8.]

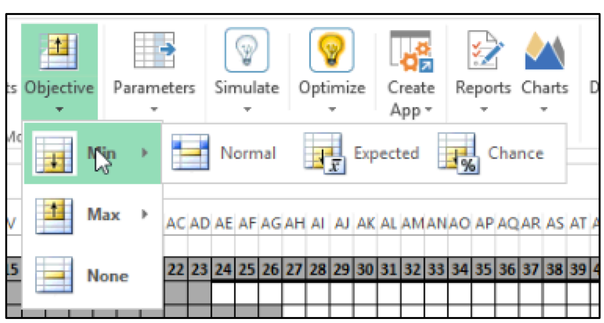


Ilustración1. 21. Ingreso de celda objetivo al programa

5. Para finalizar, se corre el programa oprimiendo la opción play . Esta opción desarrollará un análisis del ejercicio y determinará las probabilidades que se tienen para su solución. En este caso, se definió un problema tipo LP/MIP, lo que significa que es uno de los tipos de modelo más sencillos de resolver por el programa debido a la facilidad para encontrar una solución óptima global.

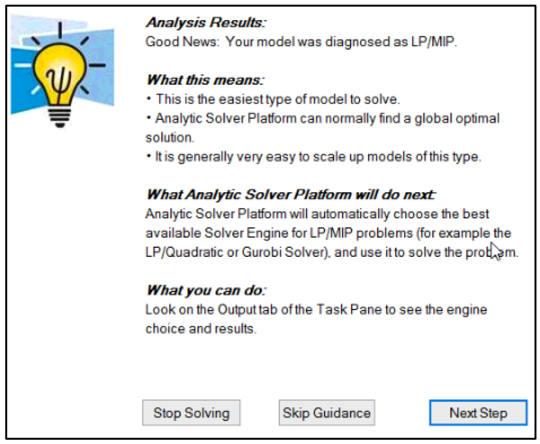


Ilustración 1.22. Resultado del análisis del programa

6. A medida que se desarrolla el problema, el programa nos muestra una estadística del progreso que se tiene a medida que se va encontrando una solución. Una vez hallada, se despliegan los resultados y el tiempo total de desarrollo.

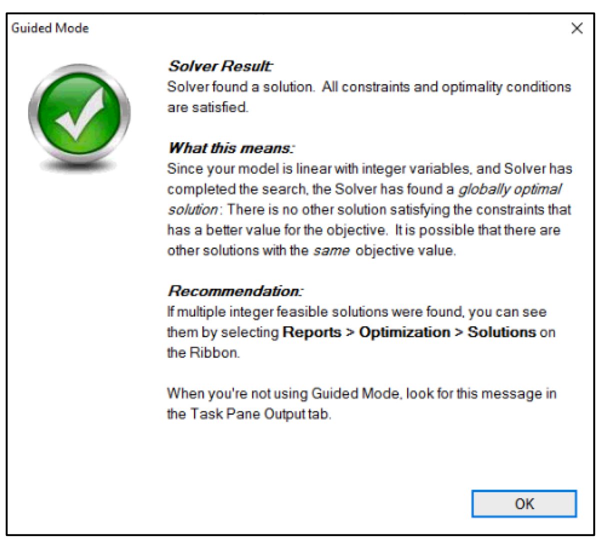


Ilustración 1.23. Mensaje de resultados de Solver Premium.

En total, el modelo se desarrolló en aproximadamente 5 minutos y 13 segundos, obteniendo un makespan final optimizado igual a 42 periodos.

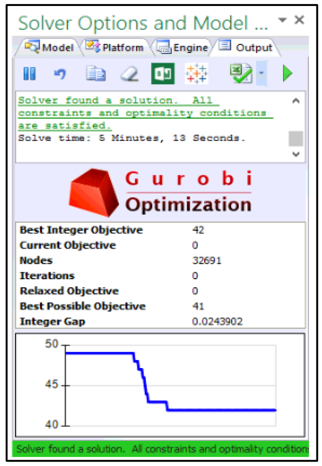


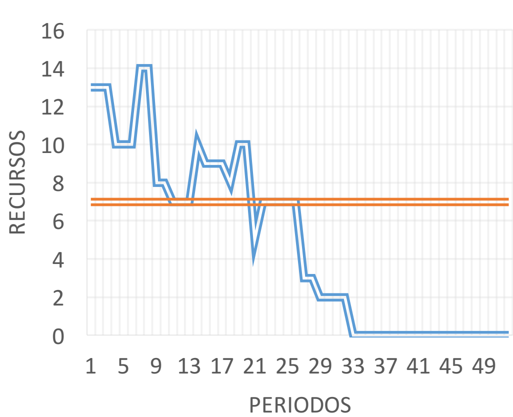
Ilustración 1.24. Procesos desarrollado por Solver para la optimización del problema.

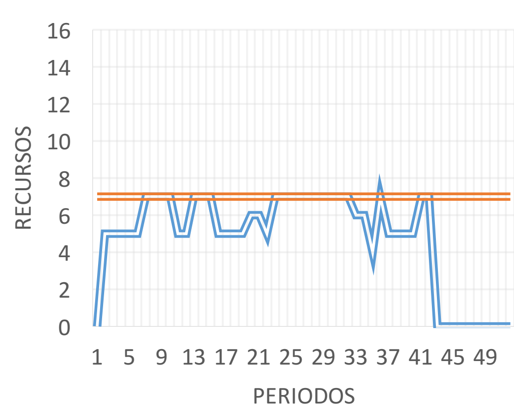
D. Resultados

Como resultado de esta programación, se incrementa el makespan del proyecto dentro de los parámetros de límites de tiempo indicados, optimizando los recursos por periodo para cada uno de los seis calendarios de recursos referenciado.

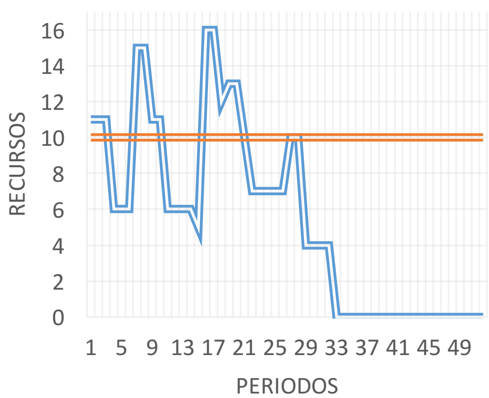
A continuación se muestra gráficamente la optimización del problema, en donde se evidencia la variación de recursos una vez el makespan aumenta dentro de los limitantes de tiempo establecidos.

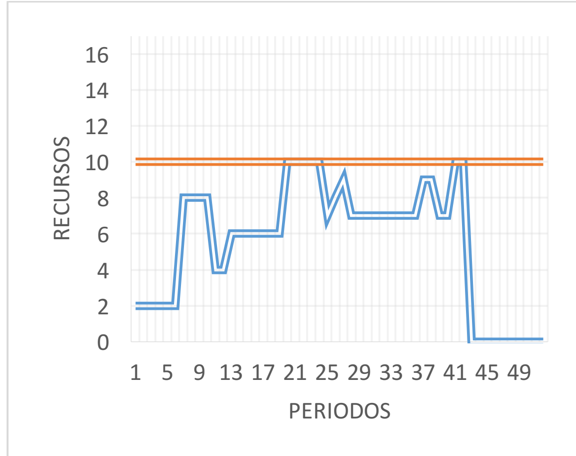
RECURSO 1	
Recurso 1 sin optimizar	Recurso 1 optimizado



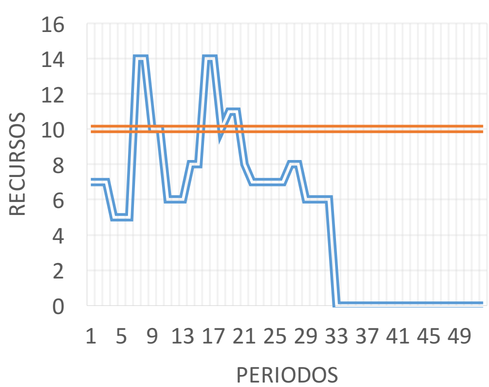


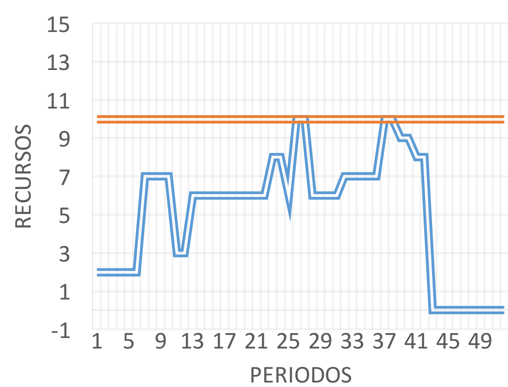
RECURSO 2	
Recurso 2 sin optimizar	Recurso 2 optimizado



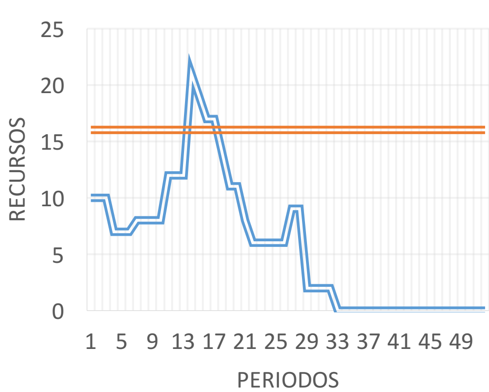


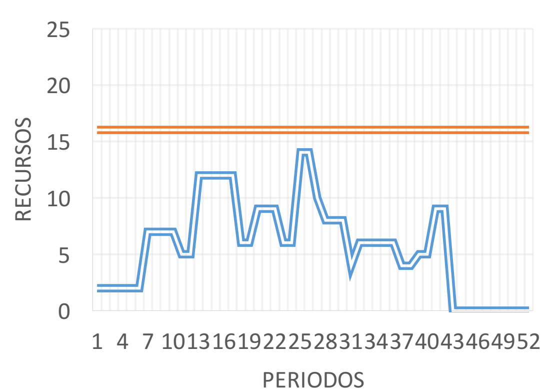
RECURSO 3	
Recurso 3 sin optimizar	Recurso 3 optimizado

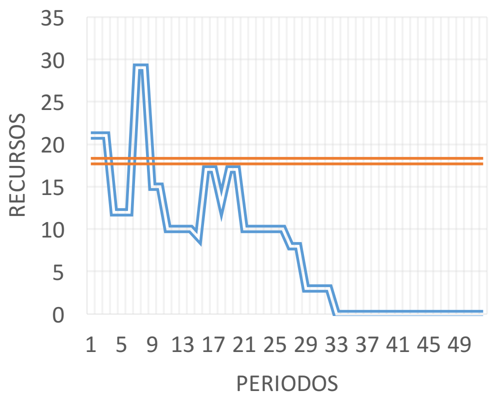
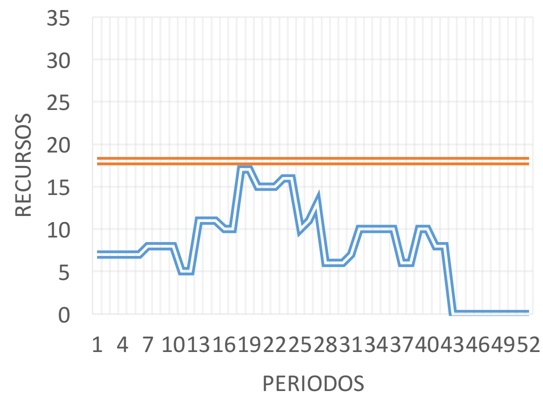


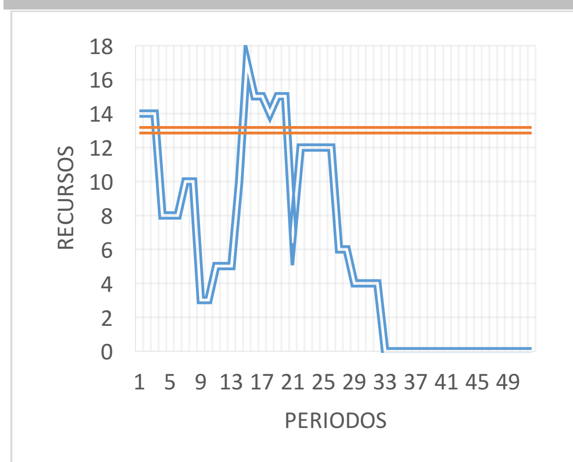
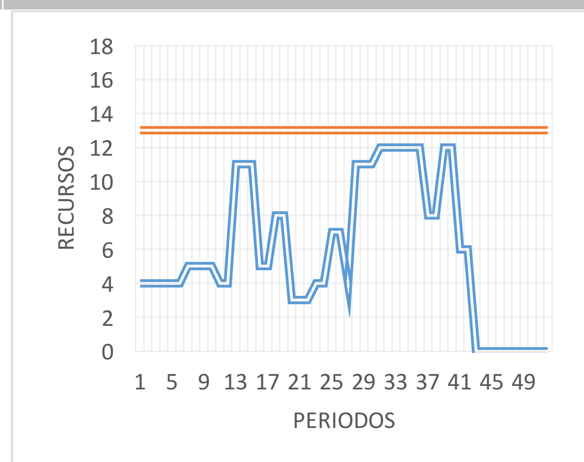


RECURSO 4	
Recurso 4 sin optimizar	Recurso 4 optimizado





RECURSO 5	
Recurso 5 sin optimizar	Recurso 5 optimizado
	

RECURSO 6	
Recurso 6 sin optimizar	Recurso 6 optimizado
	

Se puede observar en las gráficas la nivelación de recursos una vez optimizado el proyecto, esto bajo las restricciones de disponibilidad y límites de tiempo. Sin embargo, siempre es necesario aumentar el tiempo total o makespan del proyecto para que exista una solución.

Tabla 1.17. Solución final de la tabla de variables binarias.

	EF	LF	∑Xi	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	6	6	1						1																				
B	3	9	1															1											
C	10	10	1										1																
D	6	13	1												1														
E	13	16	1																										
F	15	15	1																	1									
G	8	15	1																			1							
H	8	15	1																										
I	10	17	1																										
J	21	21	1																					1					
K	14	17	1																										
L	15	19	1																										
M	18	21	1																										
N	17	21	1																										
O	18	22	1																						1				
P	26	26	1																										
Q	26	30	1																										
R	20	26	1																										
S	32	32	1																										
T	28	32	1																										
FIN	32	32	1																										

	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
A																										
B																										
C																										
D																										
E			1																							
F																										
G																										
H						1																				
I								1																		
J																										
K				1																						
L										1																
M								1																		
N										1																
O																										
P														1												
Q				1																						
R																1										
S																1										
T								1																		
FIN																1										

T	42

En este ejercicio, para un plazo máximo de 49 periodos, solo fue posible encontrar resultado viable y optimo en el periodo 42, aumentando considerablemente el makespan 10 periodos dentro de los límites de solución.

A partir de la resolución de los dos ejercicios presentados anteriormente, los cuales presentaban diferentes características de tamaño y recursos, se puede observar la gran variedad de opciones y campos en los cuales se puede aplicar la metodología de Pritsker. Esto nos permite concluir que es una herramienta de gran valor dado que promueve el perfeccionamiento de: trabajos en las diferentes áreas donde se requiera aplicar, espacios en donde se presenten inconvenientes del tipo de organización, optimización de tiempo, dinero y recursos.

2. CAPÍTULO II:
THE RESOURCE LEVELING PROBLEM (RLP)

La resolución de proyectos a partir del método de redistribución o nivelación de recursos *(The Resource Leveling Problem RLP)*, basa su teoría a partir de la necesidad de cumplir un escenario de fechas determinadas, en donde al violar los tiempos limites se puede llegar a incurrir en grandes daños si no se cumplen las condiciones de finalización establecidas.

En el ámbito de la construcción y en el entorno general, existen varios factores determinísticos que afectan el desarrollo de un proyecto los cuales pueden llegar a modificar la demanda, los recursos y algunos componentes que perturban el tiempo de ejecución. Esto ha generado que a lo largo de los años se desarrollen técnicas de programación como solución a esta problemática por medio de métodos de optimización de tiempo y recurso, los cuales permiten llegar a una adecuada gestión de proyectos de manera eficiente.

En el desarrollo de proyectos, existen cláusulas que le imponen al desarrollador o contratista el pago de determinadas multas por cada día de retraso que se presente, lo que en muchos casos puede llegar a implicar el pago de bonos y dineros extras. Por tal motivo, estas complicaciones han promovido la necesidad de desarrollar y aplicar métodos para la optimización de picos de demanda y fluctuaciones en el patrón de usos de recursos en los tiempos de trabajo (El-Rayes & Do, 2009)

Existen técnicas como PERT[4] y CPM[5] que han ayudado a grandes compañías a sobrevivir en este entorno competitivo sin lograr un gran éxito en el manejo de costos de recursos. Estos métodos fueron muy eficaces únicamente cuando no se tenía fecha límite de entrega de un proyecto y cuando no se tenían recursos limitados por cantidad o tiempo (Hegazy, 1999).

---

[4] Por sus siglas en inglés Program Evaluation and Review Technique (PERT).
[5] Por sus siglas en inglés Critical Path Method (CPM).

Como resultado, se propone la implementación de procesos prácticos de resolución de problemas que sean aplicables a grandes problemas de la vida real como RLP, capaces de encontrar una solución óptima de asignación y nivelación de recursos simultáneamente.

2.1. Contexto y antecedentes de RLP

Un gran número de autores, los cuales se mencionarán posteriormente, ya han desarrollado modelos y algoritmos enfocados a la optimización de las fluctuaciones que se presentan en la nivelación de recursos y en la reducción de los impactos negativos que se generan debido a esto. En años anteriores, se formularon métodos que buscaban la minimización de recursos a partir del desplazamiento de las actividades no críticas dentro de sus tiempos de holgura. Esto de tal forma que fuera posible mantener las duraciones totales dentro de las cotas máximas estipuladas para el proyecto, así como se presentó anteriormente en el capítulo de RCPSP.

Bajo la implementación de los mismos principios de RCPSP, ***The Resource Leveling Problem*** se busca la optimización y nivelación de recursos a partir de la suma de cuadrados de los recursos totales por periodo del proyecto. Es posible encontrar formulaciones de este método realizados en 1962 por Burgess y Killebrew, en 1976 por Ahuja, en 1978 por Harris, en 1994 por Bandelloni, en 1999 por Hegazy, en 1999 por Son, en 2013 por Ponz, entre otros. Cada uno de los autores mencionados previamente, ha aportado nuevas ideas en la aplicación del RLP en entornos un poco más realistas, como por ejemplo en la construcción.

A continuación, se presenta un modelo basado en las formulaciones propuestas por Ponz (2013), que son aplicables y programables a partir de una plantilla en Excel, y así mismo involucra la teoría anteriormente propuesta por Pritsker.

2.1.1. Formulación de Ponz (Ponz Tienda, Yepes, Pellicer, & Moreno-Flores, 2013)

La nivelación de recursos propuesta por Ponz en el año 2013, recopila los conceptos habituales del RCPSP y RLP presentando las diferencias que existen entre estos dos métodos comúnmente utilizados (Son & Skibniewski, 1999) (Leu & Hung, 2002). Cada método, busca la optimización de recursos bajo conceptos muy diferentes, el primero asume a los recursos como una limitación según las necesidades del proyecto y el segundo busca la mayor eficiencia posible de este.

El objetivo principal del RCPSP es la minimización del makespan sin exceder las restricciones de disponibilidad de recurso, mientras que los conceptos del RLP (nivelación de recursos), no buscan cumplir restricciones de disponibilidad, sino que, por el contrario, teniendo las limitaciones de tiempo establecidas, busca el rendimiento y consumo más eficiente de los recursos sin necesidad de modificar el makespan del proyecto.

Por lo tanto, la formulación propuesta pretende establecer una metodología aplicable no solo a procedimientos exactos que simplifican los problemas reales, sino también busca la obtención de soluciones optimas útiles para problemas reales del entorno.

2.1.2. Modelo

Con el fin de obtener una formulación eficiente esta teoría se basa, al igual que Pritsker, en una programación binaria y entera basada a partir de dos posibilidades de variable. La primera variable se establece en el periodo de ejecución por actividad y la segunda en el momento de finalización de cada actividad.

Para esta formulación, se optará por la segunda, por lo tanto, la variable binaria se establecerá en el momento de la finalización de una actividad, la cual está determinada de igual forma como la presentada en la ecuación 1.1 del capítulo anterior.

$$x_{it} = \begin{cases} 1 & En\ los\ casos\ donde\ una\ actividad\ i\ finaliza\ en\ el\ tiempo\ t \\ 0 & de\ lo\ contrario \end{cases}$$

siendo [2.1]

$$N = \{1, \cdots, i, i+1, \dots n\} = número\ de\ actividades$$

$$T = \{t, t+1, \dots T\} = cota\ conocida\ del\ problema\ igual\ al\ makespan$$

Con el fin de establecer las fronteras de desarrollo de cada actividad, las cuales ayudan al modelo a ser un poco más simple para su implementación en el Solver, la variable x_{it} se ejecutará únicamente cuando esta se encuentre delimitada entre los tiempos más pronto y tarde de terminar de cada actividad. Estos límites son calculados nuevamente por el método tradicional de *Forward and Backward pass.*

Al igual que en el procedimiento presentado en RCPSP, la sumatoria de los valores asumidos por la variable x_{it} dentro de los limites mencionados, debe ser igual a uno, esto a razón de su condición binaria.

$$\sum_{t=EF_i+1}^{LF_i+T-mk} x_{it} = 1 \qquad [2.2]$$

Para finalizar, se adiciona un vector de inicio programado para cada actividad a partir de la ecuación [2.2], su finalización y la duración.

$$SS_i = \sum_{t=EF_i+1}^{LF_i+T-mk} x_{it} \cdot t - d_i \qquad [2.3]$$

Función Objetivo: La función objetivo de este método busca mejorar el rendimiento y consumo de los recursos obteniendo una mejor eficiencia del proyecto, por lo tanto, no se busca la optimización y minimización de un makespan, sino que se busca nivelar los recursos a partir de la minimización de la suma cuadrática del consumo de recursos totales por cada periodo.

A continuación, se analizan las diferentes restricciones y limitaciones necesarias para lograr alcanzar el objetivo establecido.

1. **Secuenciación:** Al desarrollar un proyecto de cualquier tipo, es posible encontrar problemas de secuenciación según la dependencia de una actividad sobre otra, por lo tanto, es importante determinar actividades predecesoras y sucesoras a partir de un gráfico de nodos. Cada una de estas actividades se justifican como se presenta a continuación.

$$SS_{i-1} + d_{i-1} \leq SS_i \qquad [2.4]$$

Esto responde a que una actividad no puede iniciar hasta que su predecesora haya finalizado.

2. **Función Objetivo:** Para finalizar, se formula la ecuación objetivo, la cual está compuesta de diferentes parámetros.

I. **Sumatoria de recursos por periodo**

$$u_{tk} = \sum_{i=1}^{N} r_{ik} \qquad [2.5]$$

II. **Sumatoria cuadrática de recursos por periodo**

$$\sum_{t=1}^{T} {u_{tk}}^2 \qquad [2.6]$$

III. **Minimizar sumatoria cuadrática de todos los recursos por periodo**

$$Minimizar \sum_{k=1}^{R} \sum_{t=1}^{T} {u_{tk}}^2 \qquad [2.7]$$

Siendo

$$R = \{k, k+1, \ldots R\} = total\ recursos\ del\ problema$$

Quedando la formulación completa de la siguiente forma:

$x_{it} = \begin{cases} 1 & En\ los\ casos\ donde\ una\ actividad\ i\ finaliza\ en\ el\ tiempo\ t \\ 0 & de\ lo\ contrario \end{cases}$ $N = \{i, i+1, \ldots N\} = número\ de\ actividades$ $T = \{t, t+1, \ldots T\} = cota\ conocida\ del\ problema$	[2.1]
$\sum_{t=EF_i+1}^{LF_i+T-mk} x_{it} = 1$	[2.2]
$SS_i = \sum_{t=EF_i+1}^{LF_i+T-mk} x_{it} \cdot t - d_i$	[2.3]
$SS_{i-1} + d_{i-1} \leq SS_i$	[2.4]
$u_{tk} = \sum_{i=1}^{N} r_{ik}$	[2.5]
$\sum_{t=1}^{T} {u_{tk}}^2$	[2.6]
$Minimizar \sum_{k=1}^{R} \sum_{t=1}^{T} {u_{tk}}^2$ $R = \{k, k+1, \ldots R\} = total\ recursos\ del\ problema$	[2.7]

2.1.3. Implementación usando Solver y Solver Premium en Microsoft Excel:

Debido a la magnitud de los proyectos a los que se propone aplicar esta metodología, es recomendable la implementación de Solver Premium en Microsoft Excel, el cual nos permite desarrollar proyectos con una alta cantidad de variables sin presentar restricción alguna en el programa.

Si se desea implementar Solver de Excel es importante manejar proyectos de pequeña envergadura con el fin de que no se presenten inconvenientes al momento de encontrar una solución del método.

EJEMPLO 2.1

A. Planteamiento del problema:

Se presenta un proyecto (ver gráfico 2.1) compuesto por 10 actividades con duraciones y relaciones de precedencia establecidas.

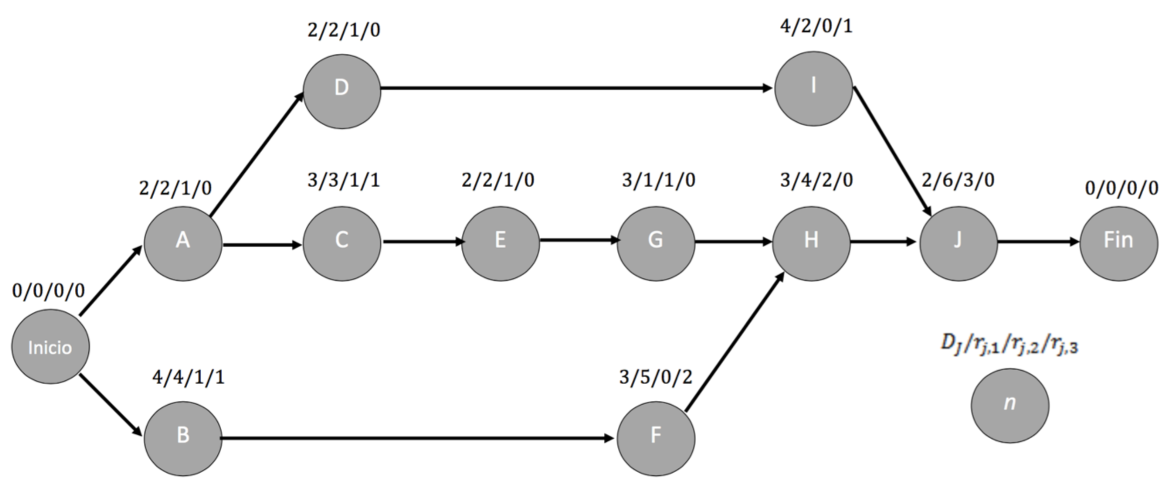


Gráfico 2.1. Gráfico de nodos y relaciones de precedencia ejemplo 2.1.

B. Solución

1. A partir de la información suministrada en el gráfico de nodos, se organiza la información y se calculan los valores de ES_i (Early Start), EF_i (Early Finish), LS_i (Late Start), LF_i (Late Finish), y el makespan del proyecto.

El cálculo del makespan del proyecto se calcula para determinar el tiempo en el cual, basados en las precedencias de las actividades, finaliza el proyecto (ecuación [1.8]). Sin embargo, el objetivo principal del problema no radica en el makespan, sino en mejorar el rendimiento de las actividades en cada periodo del proyecto a partir de la suma cuadrática de sus recursos.

Tabla 2.1. Tabla de determinación de ES, EF, LF Y LF

Actividad	Duración	R1	R2	R3	ES	EF	LS	LF
A	2	2	1	0	0	2	0	2
B	2	4	1	1	0	4	3	7
C	3	3	1	1	2	5	2	5
D	2	2	1	0	2	4	7	9
E	2	2	1	0	5	7	5	7
F	3	5	0	2	4	7	7	10
G	3	1	1	0	7	10	7	10
H	3	4	2	0	10	13	10	13
I	4	2	0	1	4	8	9	13
J	2	6	3	0	13	15	13	15
MAKESPAN		15						

2. Teniendo en cuenta que para el desarrollo del método no es necesario una cota máxima de plazo, se escoge a T=Makespan y a partir de esto se calculan los tiempos de holgura que se presentan en el proyecto de las actividades no críticas.

Tabla 2 2 Holgura máxima para el desarrollo del proyecto, ejemplo 2.1.

Mkis	15
T	15
Mkis-T	0

3. Se desarrolla la tabla principal de variables binarias en donde se establecen los periodos de holgura que existen por actividad. Estos valores se calculan al igual que en el capítulo de **RCPSP** a partir del ***Early Finish*****,** el ***Late Finish*** y aplicando el ***formato condicional 1.1***. donde se incluyen las ecuaciones [2.1] y [2.2].

Tabla 2.3 Tabla de holguras y tiempos permitidos en el que se puede desarrollar una actividad ejemplo 2.1.

Actividad	EF	LF	∑Xij	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	2	2	1		1																
B	2	7	1		1																
C	5	5	1					1													
D	4	9	1				1														
E	7	7	1							1											
F	5	10	1					1													
G	10	10	1										1								
H	13	13	1													1					
I	8	13	1								1										
J	15	15	1															1			
Fin	15	15	1															1			
Makespan	15																				

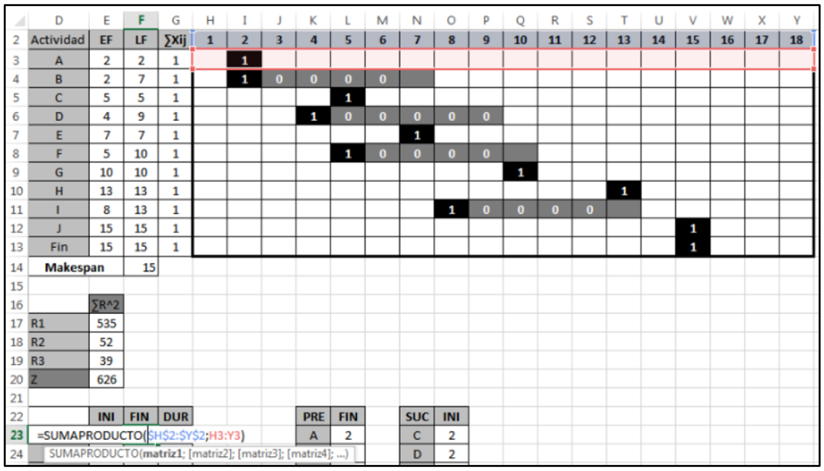
4. La Tabla 2.4 que se muestra a continuación está estrictamente relacionada con la Tabla 2.3 de holguras y variables. Se establece una relación, la cual permite visualizar los valores de inicio y fin de una actividad según como lo establece la ecuación [2.3].

Tabla 2.4. Tiempos de inicio y fin para cada actividad ejemplo 2.1.

	INI	FIN	DUR
A	0	2	2
B	0	2	2
C	2	5	3
D	2	4	2
E	5	7	2
F	7	10	3
G	7	10	3
H	10	13	3
I	9	13	4
J	13	15	2
Fin	15	15	0

Así como se especifica en la Ilustración 2.1., a partir de la herramienta ***SUMA PRODUCTO*** de Excel entre los periodos de tiempo (desde *t* hasta *T*) y las variables binarias x_{it} por actividad (desde *i* hasta *N*) se puede calcular la finalización de una tarea, mientras que su inicio se calcula a partir de la sustracción entre su finalización y su duración o la simplificación de ecuación [2.3]. El procedimiento se muestra a continuación :

$$Inicio_i = Fin_i - duración_i$$



Actividad	EF	LF	∑Xij	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	2	2	1		1																
B	2	7	1		1	0	0	0	0												
C	5	5	1					1													
D	4	9	1				1	0	0	0	0	0									
E	7	7	1							1											
F	5	10	1					1	0	0	0	0									
G	10	10	1										1								
H	13	13	1													1					
I	8	13	1								1	0	0	0	0						
J	15	15	1															1			
Fin	15	15	1															1			

Makespan	15

	∑R^2
R1	535
R2	52
R3	39
Z	626

	INI	FIN	DUR
=SUMAPRODUCTO($H$2:$Y$2;H3:Y3)			

SUMAPRODUCTO(matriz1; [matriz2]; [matriz3]; [matriz4]; ...)

PRE	FIN
A	2

SUC	INI
C	2
D	2

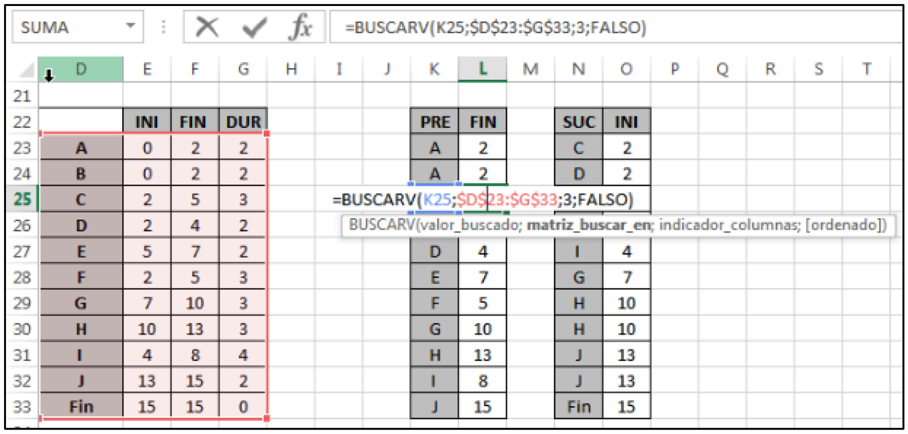
Ilustración 2.1. Parámetros para el cálculo de los tiempos de inicio- fin por actividad

5. Se definen los tiempos de inicio de una actividad sucesora con respecto a los tiempos de finalización de su actividad predecesora, estos valores se pueden obtener basados en el gráfico de precedencias o nodos y la tabla de información de tiempos por la tabla 2.4.

Tabla 2.5. Gráfico de Nodos representado de forma tabular ejemplo 2.1 .

PRE	FIN	SUC	INI
A	2	C	2
A	2	D	2
B	2	F	7
C	5	E	5
D	4	I	9
E	7	G	7
F	10	H	10
G	10	H	10
H	13	J	13
I	13	J	13
J	15	Fin	15

Como se ilustra a continuación, al utilizar la función de *BUSCARV* de Excel se simplifica considerablemente el procedimiento y se correlacionan directamente las celdas de tal forma que se modifiquen sus valores una vez ejecutada la optimización.



	INI	FIN	DUR
A	0	2	2
B	0	2	2
C	2	5	3
D	2	4	2
E	5	7	2
F	2	5	3
G	7	10	3
H	10	13	3
I	4	8	4
J	13	15	2
Fin	15	15	0

PRE	FIN
A	2
A	2
D	4
E	7
F	5
G	10
H	13
I	8
J	15

SUC	INI
C	2
D	2
I	4
G	7
H	10
H	10
J	13
J	13
Fin	15

Ilustración 2.2 Determinación de valores de precedencia ejemplo 2.1.

Se desarrolla el mismo procedimiento tanto para el cálculo del período de finalización de una actividad predecesora como para el inicio de una actividad sucesora.

6. Se desarrollan los tres calendarios de recursos correspondientes basados en la metodología utilizada en el capítulo de RCPSP y el ***FORMATO CONDICIONAL 1.2.*** Sin embargo, se presenta una variación debido a que no existen limitantes de recursos por periodo. Por lo tanto, se implementa a partir de la ecuación 2.6. una nueva celda que contenga la suma cuadrática de los recursos totales por periodo.

Tabla 2.6. Desarrollo de calendario de recursos, ejemplo 2.1.

Actividad	R1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	2	**2**	**2**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	4	**4**	**4**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	3	0	0	**3**	**3**	**3**	0	0	0	0	0	0	0	0	0	0	0	0
D	2	0	0	**2**	**2**	0	0	0	0	0	0	0	0	0	0	0	0	0
E	2	0	0	0	0	0	**2**	**2**	0	0	0	0	0	0	0	0	0	0
F	5	0	0	0	0	0	0	0	**5**	**5**	**5**	0	0	0	0	0	0	0
G	1	0	0	0	0	0	0	0	**1**	**1**	**1**	0	0	0	0	0	0	0
H	4	0	0	0	0	0	0	0	0	0	0	**4**	**4**	**4**	0	0	0	0
I	2	0	0	0	0	0	0	0	0	0	**2**	**2**	**2**	**2**	0	0	0	0
J	6	0	0	0	0	0	0	0	0	0	0	0	0	0	**6**	**6**	0	0
Fin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL	6	6	5	5	3	2	2	6	6	8	6	6	6	6	6	0	0	0
R^2	455																	

7. Para finalizar, una vez desarrollados los calendarios de recursos, se implementa a partir de la ecuación [2.7] la casilla que contiene la función objetivo (z). Esta función estará compuesta de tres valores principales, las sumas cuadráticas de los tres recursos involucrados.

Tabla 2.7. Calculo de la función objetivo.

	$\sum R^2$
R1	455
R2	52
R3	25
Z	532

C. Restricciones:

Por la simplicidad del problema que se está desarrollando, se resolverá el ejercicio a partir del Solver integrado en Excel. Para encontrar la optimización de este problema es necesario la implementación de las siguientes restricciones:

1. Establecer aquellas celdas que son variables ubicadas en la tabla inicial 2.3. de variables binarias.

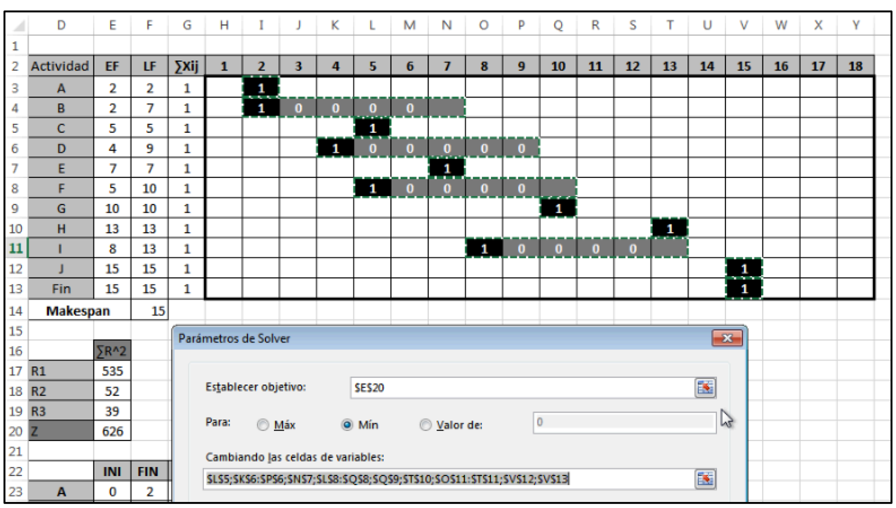
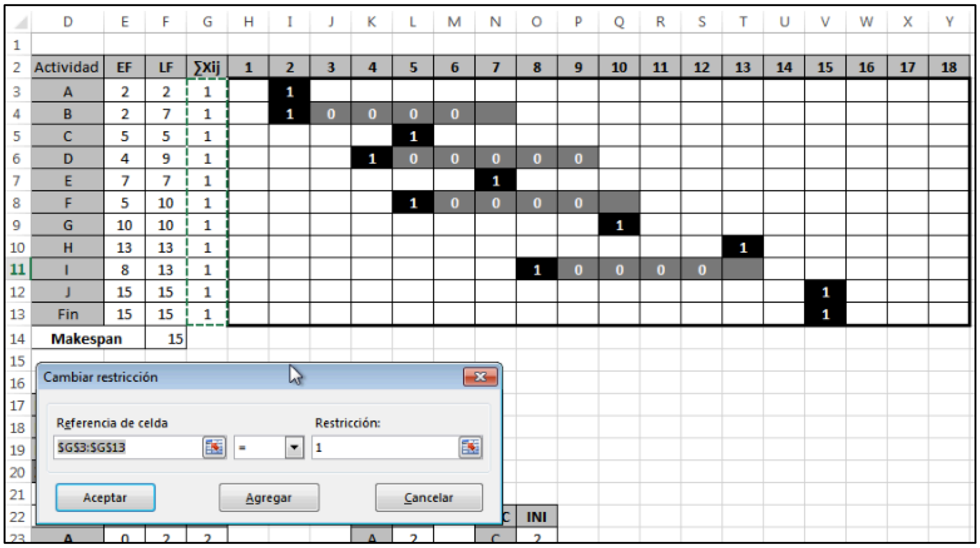


Ilustración 2.3. Selección de variables binarias en Excel.

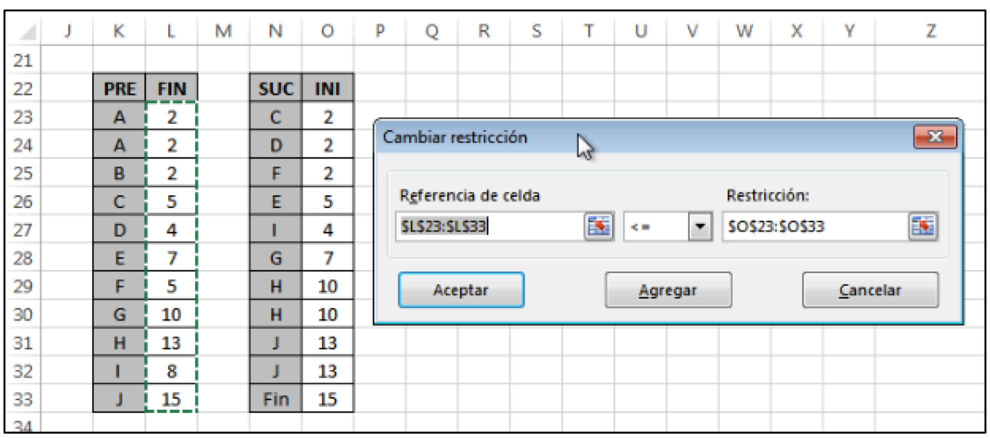
2. La sumatoria *ΣXit* por actividad de las variables binarias por periodo deben ser estrictamente uno.



Actividad	EF	LF	ΣXij	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	2	2	1		1																
B	2	7	1		1	0	0	0	0												
C	5	5	1					1													
D	4	9	1				1	0	0	0	0	0									
E	7	7	1							1											
F	5	10	1					1	0	0	0	0									
G	10	10	1										1								
H	13	13	1													1					
I	8	13	1								1	0	0	0	0						
J	15	15	1															1			
Fin	15	15	1															1			
Makespan		15																			

Ilustración 2.4. Selección de sumatoria de periodos por actividad asignando restricción de ΣXit=1, ejemplo 2.1.

3. Se ingresan las restricciones de precedencia y secuenciación de los valores de inicio y fin representadas en la tabla 2.5.



PRE	FIN	SUC	INI
A	2	C	2
A	2	D	2
B	2	F	2
C	5	E	5
D	4	I	4
E	7	G	7
F	5	H	10
G	10	H	10
H	13	J	13
I	8	J	13
J	15	Fin	15

Ilustración 2.5. Ingreso de Restricciones de precedencia al programa

4. Se seleccionan aquellas variables resaltadas en el *formato condicional 1.1.* en una restricción binaria haciendo validez a lo establecido en la ecuación [2.1].

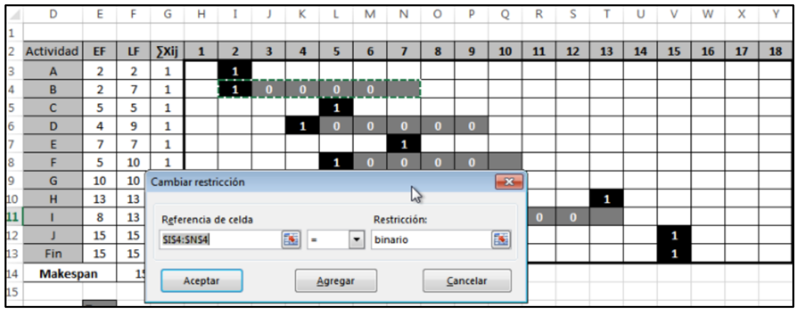


Ilustración 2.6. Ingreso de restricción de variables binarias al problema ejemplo 2.1.

5. Se señala la casilla que contiene a la función objetivo del problema ecuación [2.7].

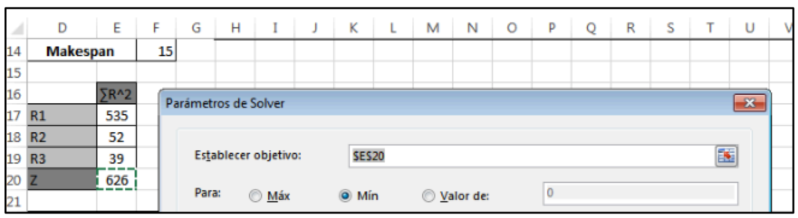


Ilustración 2.7. Selección función objetivo del problema ejemplo 2.1.

6. Por último, se desarrolla la plantilla anteriormente programada en Excel a partir de la opción ***EVOLUTIONARY***, esto se debe a que a pesar de que se está programando de forma lineal se están implementando funciones cuadráticas, que para su solución no sería posible utilizar el método de resolución ***SIMPLEX*** debido a que no encontraría un procedimiento óptimo para dar respuesta del problema.

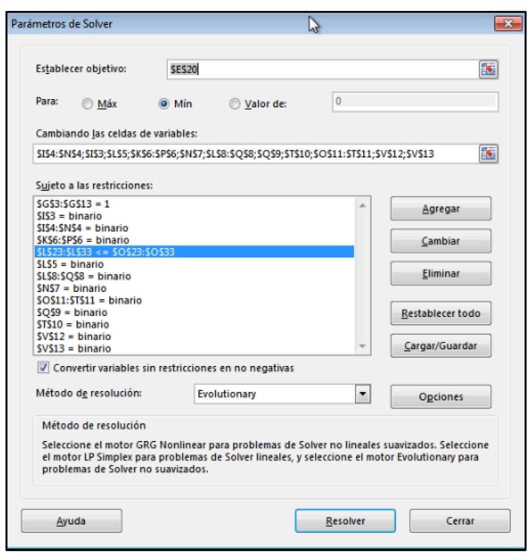


Ilustración 2.8. Método de resolución para los parámetros de Solver de Excel. Ejemplo 2.1.

D. Resultados

Como resultado de la programación, se espera minimizar la función objetivo z del proyecto de tal forma que sea posible obtener una nueva distribución de recursos según la reorganización de las tareas que lo componen.

Inicialmente, si las actividades se desarrollan consecutivamente sin la necesidad de utilizar los tiempos de holgura se obtiene un z total de 626, valor que se desea reducir. Una vez se optimiza, es posible obtener un z final de 514 obteniendo una nueva nivelación de los recursos.

Tabla 2.8. Valores de la celda objetivo antes y después de optimizar, ejemplo 2.1.

	$\sum R^2$ **Inicial**	$\sum R^2$ **Final**
R1	535	435
R2	52	52
R3	39	27
Z	626	514

A continuación, se pueden observar los cambios presentados en la sumatoria total de los recursos utilizados por periodo en cada uno de los calendarios de recurso desarrollado. Es importante resaltar que una vez la optimización se ejecuta, los picos y la acumulación de recursos que se presentan inicialmente se redistribuyen de tal forma que se mejora la eficiencia del proyecto.

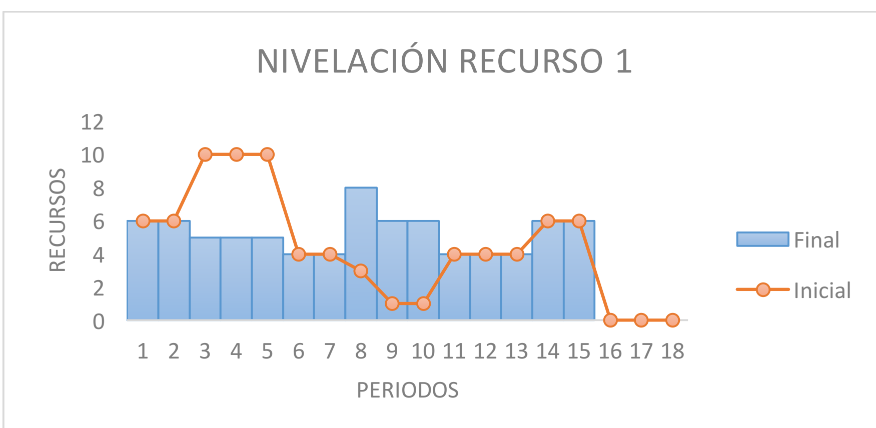


Gráfico 2.2. Cambios en la nivelación por periodos del recurso 1, ejemplo 2.1.

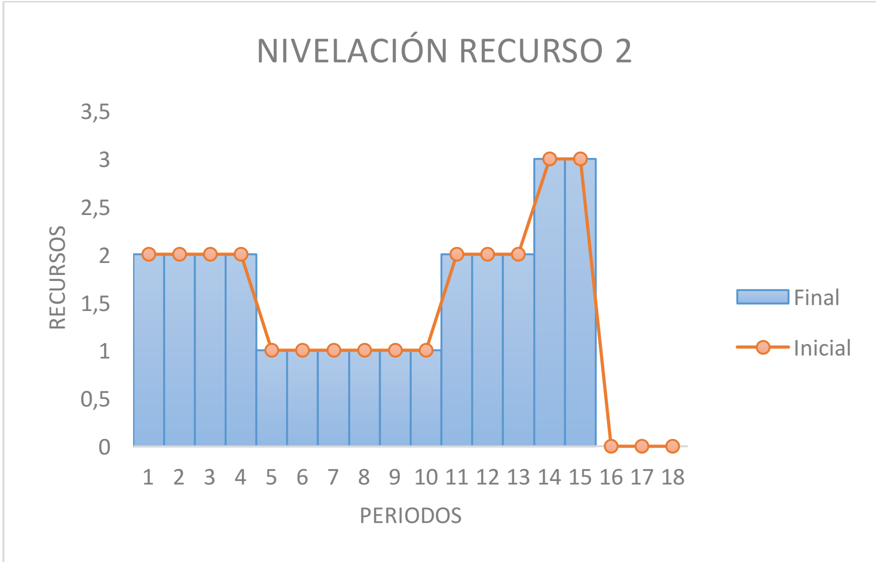


Gráfico 2.3. Cambios en la nivelación por periodos del recurso2, ejemplo 2.1

Gráfico 2.4. Cambios en la nivelación por periodos del recurso 3, ejemplo 2.1

A continuación, se presenta la nueva organización optimizada de las tareas en donde se cumplen cada una de las restricciones.

Tabla 2.9. Resultado final nivelación de recursos.

Actividad	EF	LF	Σ Xij	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	2	2	1		1																
B	2	7	1		1	0	0	0	0	0											
C	5	5	1					1													
D	4	9	1				1	0	0	0	0	0									
E	7	7	1							1											
F	5	10	1					0	0	0	0	0	1								
G	10	10	1										1								
H	13	13	1													1					
I	8	13	1								1	0	0	0	0	0					
J	15	15	1															1			
Fin	15	15	1															1			
Makespan																		15			

A partir de la tabla 2.9, se puede observar que únicamente se presenta variación entre las tablas 2.3 y 2.9 en el momento de procesar la actividad F. Esto demuestra, que retardar al menos la ejecución de una actividad dentro de las fronteras de holgura, puede afectar considerablemente la eficiencia de un proyecto, y por lo tanto se puede afirmar que se justifica la implementación del método para el desarrollo de este tipo de ejercicios.

Con el fin implementar el método RLP en problemas de mayor magnitud de actividades y recursos, se propone la resolución nuevamente del ejemplo 1.2 del capítulo anterior.

EJEMPLO 2.2 (IMPLEMENTACIÓN SOLVER PREMIUM)

A. Planteamiento del problema:

A partir de la solución del ejemplo 1.2. del capítulo de RCPSP, se plantea resolver nuevamente el ejercicio con T=Makespan utilizando el método de ***Resource Leveling Problem***.

B. Solución

1. A partir de los datos calculados en la tabla 1.12., se desarrolla nuevamente la tabla de variables binarias x_{it} renovando únicamente los tiempos de holgura que se tienen por actividad a partir de T=Makespan.

Tabla 2.10. Holgura máxima para el desarrollo del proyecto, ejemplo 2.2.

Makespan	32
T	32
Makes-T	0

Tabla 2.11. Tabla de holguras y tiempos permitidos en el que se puede desarrollar una actividad, ejemplo 2.2.

	EF	LF	∑Xij	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
A	6	6	1						1																											
B	3	9	1			1																														
C	10	10	1										1																							
D	6	13	1						1																											
E	13	16	1													1																				
F	15	15	1															1																		
G	8	15	1								1																									
H	8	15	1								1																									
I	10	17	1										1																							
J	21	21	1																					1												
K	14	17	1														1																			
L	15	19	1															1																		
M	18	21	1																		1															
N	17	21	1																	1																
O	18	22	1																		1															
P	26	26	1																										1							
Q	26	30	1																										1							
R	20	26	1																				1													
S	32	32	1																																1	
T	28	32	1																												1					
FIN	32	32	1																																1	

2. Se utiliza nuevamente la tabla de precedencias (ver tabla 1.15) en donde se determinan los periodos de finalización y de inicio de las actividades predecesoras y sucesoras para la implementación de restricciones.
3. Se desarrollan nuevamente los seis calendarios de recursos implementando los cambios necesarios que implica el método RLP, por lo tanto, se añaden nuevas casillas que contengan la suma cuadrática de recursos.

Tabla 2.12. Desarrollo de calendario de recursos, ejemplo 2.2.

	R1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
A	5	5	5	5	5	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	2	0	0	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	5	5	5	5	5	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	3	0	0	0	0	0	0	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	4	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	4	0	0	0	0	0	0	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	5	0	0	0	0	0	0	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	3	0	0	0	0	0	0	0	0	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
K	3	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	3	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	3	3	0	0	0	0	0	0	0
Q	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4	4	4	0	0	0	0	0	0	0
R	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0
S	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	0
T	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
SUMA		13	13	13	10	10	10	14	14	8	8	7	7	7	10	9	9	9	8	10	10	5	7	7	7	7	7	3	3	2	2	2	2	0
SUMA R^2	2385																																	

4. Por último, se establece la función objetivo z a partir de la sumatoria de los resultados de R^2 en cada uno de los calendarios.

Tabla 2.13. Desarrollo celda función objetivo, ejemplo 2.2.

	$\sum R^2$
R1	2385
R2	2935
R3	2365
R4	3375
R5	6457
R6	3463
Z	**20980**

Los valores que se presentan anteriormente se encuentra sin optimizar y se obtienen cuando las actividades se desarrollan en secuencia sin utilizar los tiempos de holgura que se presenten para las actividades no críticas.

C. Restricciones (Implementación Solver Premium):

Teniendo en cuenta la cantidad de recursos que se está utilizando es recomendable desarrollar el método por medio de Solver Premium, sin embargo el límite de celdas variables que se están manipulando permite el uso de Solver de Microsoft Excel.

Para el desarrollo del ejercicio, al igual que en el ejemplo 1.2, se utiliza la herramienta ***ANALYTIC SOLVER PLATAFORM*** instalada previamente y se efectúan las nuevas restricciones.

1. Debido a los cambios en los tiempos de holgura es necesario seleccionar en el programa nuevamente las celdas que son variables con la función *Add Normal Variable.*

2. Se ingresan nuevamente las restricciones del problema debido a que se ha modificado la función objetivo y se desea optimizar los recursos a partir de su suma cuadrática.
Por tal motivo, se ingresan las siguientes restricciones:

  2.1. La suma cuadrática de las variables debe ser estrictamente igual a uno debido a su condición binaria, (ver ecuación [2.1]).

  2.2. Basándose en la tabla 1.15, se establecen las restricciones de precedencia donde el periodo de finalización de una tarea predecesora debe ser menor o igual al inicio de su tarea sucesora.

  2.3. Es necesario establecer las características de las celdas variables como binarias

3. Por último, se escoge la nueva función objetivo (z) la cual está determinada por la suma total de las cuadráticas de los recursos por periodo en la tabla 2.13.

4. Una vez se corre el programa el análisis nos arroja que es un problema tipo QP/MIP el cual es posible resolver a partir de las diferentes maquinas que nos ofrece el programa.

Para este ejemplo particular, el programa establece que se puede utilizar ***The LP/Quadratic engine.***

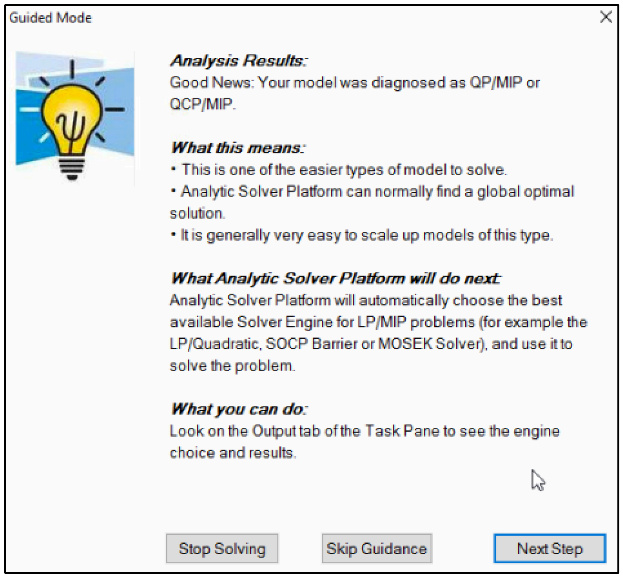


Ilustración 2.9. Resultado del análisis del programa, ejemplo 2.2.

5. Una vez solucionado el problema se puede observar el procedimiento que el programa ha realizado para encontrar el resultado más óptimo, el cual satisface todas las restricciones establecidas y minimiza en lo posible la función objetivo.

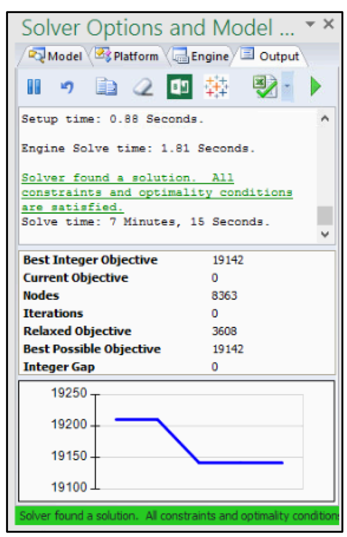


Ilustración 2 10. Proceso desarrollado por Solver para la optimización del problema, ejemplo 2.2.

D. Resultados

Como resultado de la programación, se obtiene una nueva organización en el desarrollo de las actividades del proyecto, nivelando los picos y recursos por periodo, lo cual genera una disminución considerable de la sumatoria cuadrática.

Tabla 2.14. Valores de la celda objetivo antes y después de optimizar, ejemplo 2.2.

	$\sum R^2$ Inicial	$\sum R^2$ Final
R1	2385	2209
R2	2935	2665
R3	2365	2283
R4	3375	3169
R5	6457	5605
R6	3463	3211
Z	**20980**	**19142**

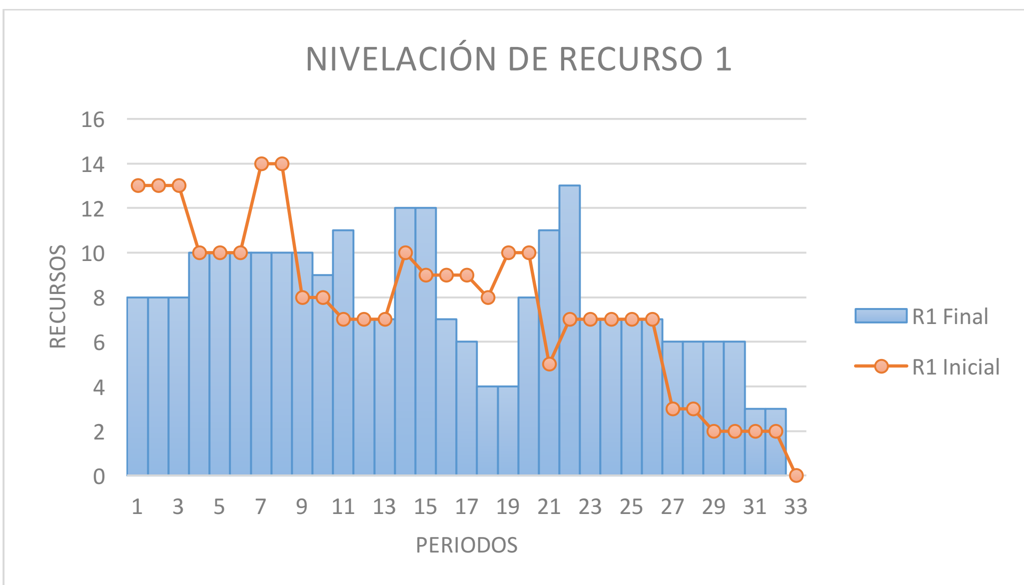


Gráfico 2.5. Cambios en la nivelación por periodos del recurso 1, ejemplo 2.2.

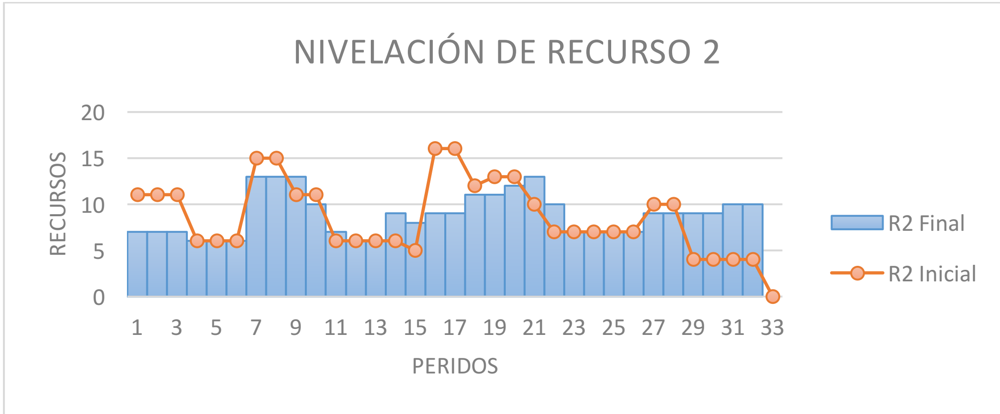


Gráfico 2.6. Cambios en la nivelación por periodos del recurso 2, ejemplo 2.2.

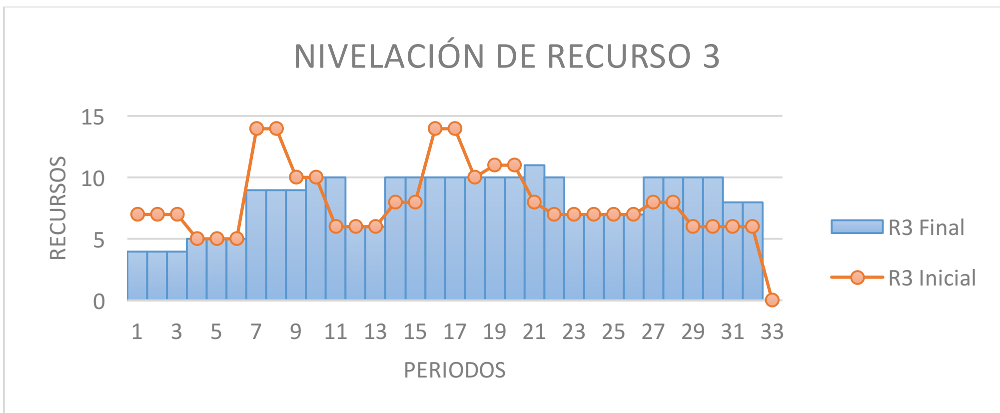


Gráfico 2.7. Cambios en la nivelación por periodos del recurso 3, ejemplo 2.2.

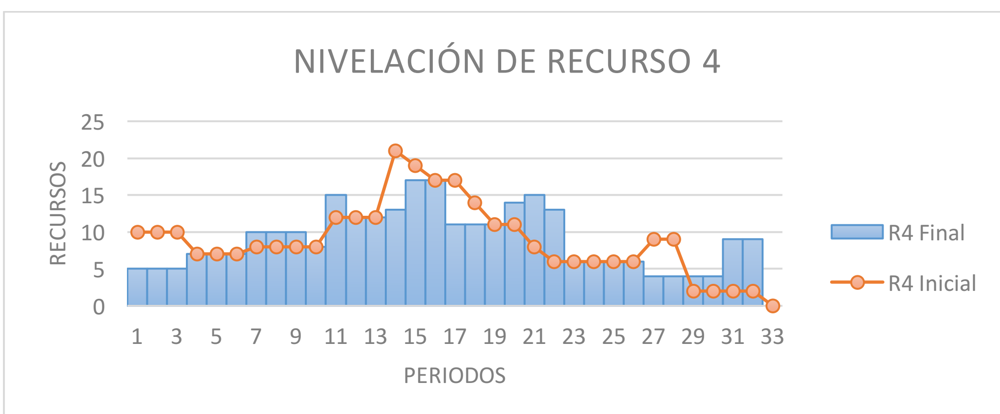


Gráfico 2.8. Cambios en la nivelación por periodos del recurso 4, ejemplo 2.2.

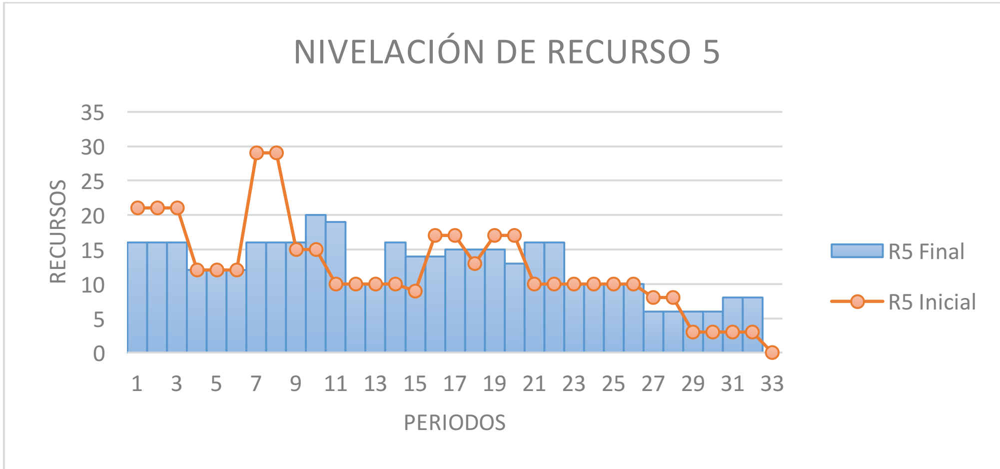


Gráfico 2.9. Cambios en la nivelación por periodos del recurso 5, ejemplo 2.2.

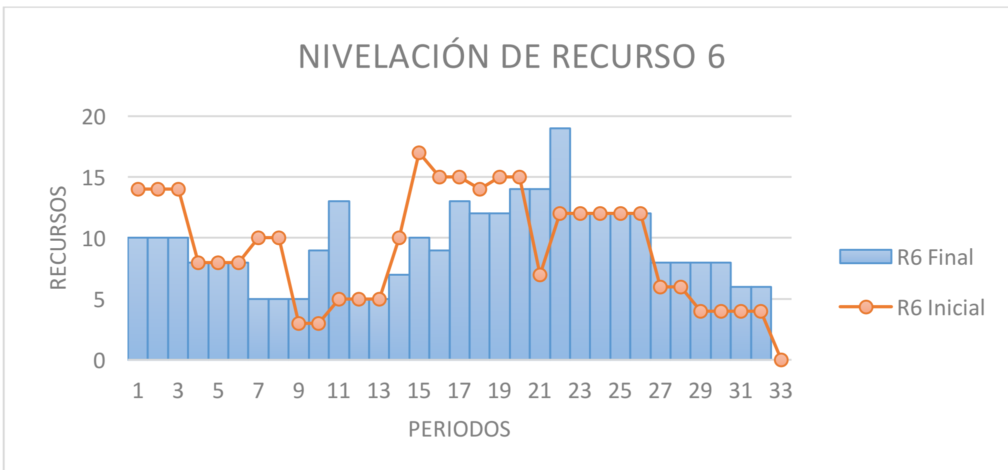


Gráfico 2.10. Cambios en la nivelación por periodos del recurso 6, ejemplo 2.2.

En conclusión, la implementación de métodos como RLP le permitirá al gerente y organizador de proyectos maximizar la eficiencia del manejo de sus recursos por periodo, y esto sin importar el ámbito de trabajo al cual se esté implementando y sin modificar el makespan del proyecto.

3. Conclusiones

Este tipo de modelos prácticos programables en Microsoft Excel, están diseñados para que exista un calendario de ejecución de tareas con un mínimo de fluctuaciones de recursos en donde los picos de demandas se minimicen efectivamente cumpliendo el objetivo establecido a partir del método utilizado, minimizar tiempos (RCPSP) o aumentar la eficacia (RLP). Así mismo, a lo largo del desarrollo de los métodos, son posibles las siguientes conclusiones.

- Primero calcular la efectividad del cronograma inicial del problema, en donde se muestran los tiempos de holgura de aquellas actividades que se pueden retardar o agilizar sin afectar la continuidad de los procesos ni el makespan final.

- Segundo, al finalizar cualquiera de los métodos estudiados se obtiene un nuevo cronograma de actividades que optimiza el procesamiento de los recursos optimizando tiempo o aumentando la eficacia según sea el caso de estudio.

- Por último, se puede observar que el retraso en la ejecución de cualquier actividad puede afectar considerablemente el desarrollo óptimo del proyecto, especialmente en los casos en donde está actividad hace parte de la ruta crítica del proyecto, por lo tanto, como gerente de proyectos es vital el cumplimiento de cronogramas para el éxito del proyecto.

BIBLIOGRAFÍA

Pritsker, A. W. (1969). Multiproject scheduling with limited resources A Zero-One programming approach. USA: INFORMS.

Álvarez-Valdés, R., & Tamarit. (1993). The Project Scheduling Polyhedron: Dymension, Facets and Lifting Theorems. *European Journal of Operation Research.*

Mingozzi A, V. M. (1998). *An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation.*

Klein, R. (2000). *Scheduling of Resource-Constrained Projects.* (Kluwer, Ed.) Boston.

Proon, S. J. (2010). *A Genetic Algorithm with Neighborhood Search for the Resource-Constrained Project Scheduling Problem.* Pittsburgh, Mississippi: Wiley Online Library.

Artigues, C. D. (2008). *Resource-Constrained Project Scheduling. Models, Algorithms, Extensions and Aplications.* (I. ISTE Ltd and John Wiley & Sons, Ed.) USA.

Weglarz, J. (1999). *Project Scheduling, Recent Models, Algorithms and Applications.* Standford, California, USA: Kluwer Academic Publishers.

Pinha, D. A. (2015). Parallel Mode Schedule Generation Scheme.

Demeulemeester, E. H. (2002). *Project Scheduling, A Research Handbook.* USA: Kluwer Academic Publishers.

El-Rayes, K., & Do, H. (2009). *Optimizing Resource Leveling in Construction Projects.* ASCE.

Hegazy, T. (1999, Junio). OPTIMIZATION OF RESOURCE ALLOCATION AND LEVELING USING GENETIC ALGORITHMS. *JOURNAL OF CONSTRUCTION ENGINEERING AND MANAGEMENT*.

Ponz Tienda, J. L., Yepes, V., Pellicer, E., & Moreno-Flores, J. (2013). The Resource Leveling Problem with multiple resources using an adaptive genetic algorithm. *Automation in Construction*. Valencia, España: Elsevier B.V. .

Son, J., & Skibniewski, M. (1999, Enero). MULTIHEURISTIC APPROACH FOR RESOURCE LEVELING PROBLEM IN CONSTRUCTION ENGINEERING: HYBRID APPROACH. *JOURNAL OF CONSTRUCTION ENGINEERING AND MANAGEMENT*.

Leu, S.-S., & Hung, T.-H. (2002). An optimal construction resource leveling scheduling simulation model. *Can. J. Civ. Eng. , 29*, 267-275.

país.com.co, E. (2012). *Mala planeación retrasa las grandes obras de infraestructura en Colombia.* Retrieved from El país: http://www.elpais.com.co/elpais/economia/noticias/mala-planeacion-retrasa-grandes-obras-infraestructura-en-colombia