



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

*OPTIMIZACIÓN EN EL PROCESO DE GESTIÓN DE PROYECTOS DE  
CONSULTORÍA DE INGENIERÍA*

**PROYECTO DE GRADO PARA OPTAR AL GRADO DE MAGÍSTER EN  
INGENIERÍA DE NEGOCIOS CON TECNOLOGÍAS DE INFORMACIÓN**

**ALEJANDRO PATRICIO TEIXEIRA GOMEZ**

**PROFESOR GUÍA:**

OSCAR BARROS VERA

**MIEMBROS DE LA COMISIÓN:**

PATRICIO ANGUIA OSORIO

LUIS SERPA TAMAYO

SANTIAGO DE CHILE

DICIEMBRE 2012

## RESUMEN

Controlca S.A., es una empresa venezolana, fundada en los años 70, con el fin de ofrecer servicios de consultoría en el área de las telecomunicaciones. Controlca tiene una amplia trayectoria en este ámbito, lo cual le ha permitido posicionarse entre las más prestigiosas consultoras de Venezuela. Si bien, es cierto que Controlca tiene esta trayectoria, requiere mejoras en su sistema de gestión y control, específicamente en el manejo y control de horas hombre llevadas a cabo por los líderes de proyecto y demás ingenieros involucrados en las tareas de diversos proyectos que la empresa ejecuta.

Específicamente, la necesidad de la empresa es conseguir un manejo eficiente de sus proyectos y de los recursos asignados a los mismos. Asimismo, requiere de la optimización de la asignación de tareas al recurso humano capacitado, en base al perfil de conocimiento que poseen.

De allí entonces, surge la motivación de desarrollar una solución que permita gestionar y planificar eficientemente la utilización de las horas hombre para la realización de proyectos. La solución se generará desde el punto de vista interno de la empresa, para dar una respuesta más acertada y cercana a la realidad. Lo anterior permitirá conducir el proyecto de manera que esta solución sea posible y aplicable en un lapso mediano de tiempo. El impacto que tendrá esta solución estará relacionado directamente con su principal cliente Inelectra, en términos de eficiencia y de entregar productos de alta calidad a menores costos y en el menor tiempo posible.

De esta manera, se da inicio a este Proyecto de Grado "Optimización en el proceso de gestión de proyectos de consultoría de ingeniería". El mismo hace énfasis en el rediseño de los procesos internos de Controlca, con el fin de incorporar nuevos procesos y tecnologías. Busca en fin, mejorar la planificación, gestión, administración y seguimiento de todos sus proyectos de telecomunicaciones de forma óptima y eficiente. Se alcanzarán entonces mejores resultados, tangibles y medibles, tanto para la empresa como para su principal cliente, Inelectra.

## **ABSTRACT**

Controlca S.A is a Venezuelan company founded in the 70s to provide consulting services in the area of telecommunications. Controlca has an extensive experience in this area, which has positioned it among the most prestigious consulting firms in Venezuela. Although it's true that Controlca has this many years of experience, it also requires improvements in its management and control system, specifically in the management and control of man-hours performed by project leaders and other engineers involved in the tasks of various projects that the company carries out.

More precisely, the need for the company is to efficiently manage its projects and the resources assigned to them. It also requires the optimization of the assignment of tasks to trained human resources based on the knowledge profile they have.

Because of this, the motivation to develop a solution that efficiently manages and plans the use of man-hours for the completion of projects arises. The solution will be generated from the internal point of view of the company to give a more accurate and closer to reality response. This will allow to lead the project in a way that this solution will be feasible and applicable within a medium period of time. The impact of this solution will be directly related to its main customer Inelectra, in terms of efficiency and delivering of high quality products at lower costs and in the shortest time possible.

Thus the Graduation Project "Optimization in the project management process engineering consulting" begins. This project emphasizes on redesigning Controlca's internal processes in order to incorporate new processes and technologies. It seeks to improve the planning, management, administration and monitoring of all telecommunication projects in an optimal and efficient way. Then, better, tangible and measureable results will be achieved, both for the company and its main customer, Inelectra.

## DEDICATORIA

*A Dios.*

*Por haberme dado salud y fuerza para lograr este anhelado objetivo profesional. Por su infinita bondad y amor.*

*A mis padres.*

*Por ser el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como de la vida, por su incondicional apoyo perfectamente mantenido a través del tiempo.*

*A mi amada esposa.*

*Por su incansable apoyo, dedicación y amor que me brinda día a día para alcanzar nuevas metas, tanto profesionales como personales.*

## **AGRADECIMIENTOS**

*A mis amigos.*

*Quienes me han apoyado durante todos estos años en mi formación profesional y que hasta ahora, seguimos siendo amigos: Eduardo Ferro, Gustavo Angulo, José Enrique González Cornejo, Luis Eduardo Serpa, Josefina Tamayo y Eduardo Stiassni.*

*A Ana María, Laura y Don Oscar por su total aceptación y disposición en este proceso.*

## INDICE

1. PLANTEAMIENTO Y MOTIVACIONES DEL PROYECTO.....	1
2. ANTECEDENTES DE LA INDUSTRIA Y LA EMPRESA.....	2
2.1. La Empresa: Controlca.....	2
2.2. Alianza estratégica.....	5
2.3. Misión y Visión de Controlca.....	5
2.3.1. Misión.....	5
2.3.2. Visión.....	6
2.4. Objetivos.....	6
2.5. Estructura organizativa.....	6
2.6. Clientes.....	7
2.7. INELECTRA S.A.C.A. ....	8
3. MARCO TEORICO CONCEPTUAL.....	9
3.1. RCPSPP: Resource-Constrained Project Scheduling Problem.....	9
3.1.1. Definición del problema.....	11
3.1.2. Metodología.....	13
3.1.3. RESCON.....	16
3.2. Marco Teórico Conceptual: PMBoK.....	16
3.2.1. Definiciones.....	18
3.2.2. Ciclo de vida de un proyecto.....	19
3.2.3. Interesados.....	20
3.2.4. Descripción de los procesos por etapa.....	21
3.2.5. Agrupamiento de Procesos por Área de Gestión.....	26
4. ANALISIS DEL MEDIO.....	26
4.1. Análisis de las Fuerzas de Porter.....	27
4.1.1. Poder de negociación de los Compradores o Clientes.....	27
4.1.2. Poder de negociación de los Proveedores o Vendedores.....	28
4.1.3. Amenaza de nuevos entrantes.....	28
4.1.4. Amenaza de productos sustitutos.....	29
4.1.5. Rivalidad entre los competidores.....	30

4.2.	Análisis FODA.....	30
4.2.1.	Fortalezas.....	30
4.2.2.	Oportunidades.....	30
4.2.3.	Debilidades.....	31
4.2.4.	Amenazas.....	31
4.3.	Caracterización del mercado.....	31
4.3.1.	Estado actual.....	31
4.3.2.	Tamaño.....	32
4.3.3.	Segmentación.....	32
4.3.4.	Localización.....	32
5.	EL PROYECTO.....	33
5.1.	Planteamiento Estratégico.....	33
5.1.1.	Planteamiento estratégico, tipo de diferenciación y ventajas competitivas.....	35
5.1.2.	Posicionamiento Estratégico según Hax.....	36
5.1.3.	Solución integral al cliente.....	37
5.2.	Modelo de Negocios.....	39
5.3.	Propuesta de valor.....	41
5.3.1.	Variables a considerar.....	41
5.4.	Recursos Claves.....	42
5.5.	Procesos Claves.....	43
5.6.	Beneficios esperados.....	43
5.6.1.	Cuantificación de los Beneficios.....	44
5.7.	Definición del Proyecto.....	44
5.7.1.	El problema.....	45
5.7.2.	Objetivos estratégicos.....	45
5.7.3.	Evaluación Financiera.....	47
5.7.4.	Plan de Marketing.....	48
6.	REDISEÑO DEL PROCESO DE NEGOCIOS Y APOYO COMPUTACIONAL.....	49
6.1.	Relación con el cliente.....	53
6.2.	Planificación, Programación y Control Proyectos de Telecomunicaciones.....	57
6.3.	Control de Proyectos.....	62
7.	DISEÑO IMPLEMENTADO COMO APOYO A LOS PROCESOS.....	64

7.1.	DIAGRAMA DE PAQUETES.....	64
7.2.	CASOS DE USO.....	66
7.2.1.	Detalles de los Casos de Uso.....	67
7.2.2.	Ingreso Proyectos.....	67
7.2.3.	Ingreso Tareas.....	68
7.2.4.	Buscar Proyectos Similares.....	68
7.2.5.	Buscar Proyectos.....	68
7.2.6.	Asignación Tareas.....	68
7.2.7.	Ver Recursos Disponibles.....	69
7.2.8.	Ingreso Recursos.....	69
7.2.9.	Notificar Asignaciones.....	69
7.2.10.	Control de Tareas.....	69
7.2.11.	Registro de Tareas.....	70
7.2.12.	Informe de Avances.....	70
7.3.	DIAGRAMAS DE ESCENARIOS.....	70
7.3.1.	Ingreso de Proyectos.....	71
7.3.2.	Planificación de Proyectos.....	72
7.3.3.	Control de Proyectos.....	72
7.4.	DIAGRAMAS DE SECUENCIA EXTENDIDOS Y CLASES.....	74
7.4.1.	Ingreso de Proyectos.....	75
7.4.2.	Planificación de Proyectos.....	79
7.4.3.	Control de Proyectos.....	82
8.	PRUEBA DE CONCEPTO.....	85
9.	DESARROLLO DEL PROTOTIPO.....	95
9.1.1.	Control de Proyectos.....	96
9.1.2.	Listado y Creación de Proyectos.....	97
9.1.3.	Planificación de Proyectos.....	99
9.1.4.	Sistema de Alertas.....	101
9.1.5.	Actualización de Proyectos.....	102
9.1.6.	Autogeneración de cartas Gantt.....	102
9.1.7.	Panel de Control.....	103
10.	GENERALIZACIÓN DE LA EXPERIENCIA.....	105



11.	PERSPECTIVAS FUTURAS DEL PROYECTO DE GRADO .....	108
12.	BIBLIOGRAFIA.....	110

## **1. PLANTEAMIENTO Y MOTIVACIONES DEL PROYECTO**

A finales de la década de los 70's Controlca inicia operaciones en el área de consultoría, para dar respuesta a las demandas en el rubro de las telecomunicaciones.

Ya han pasado más de treinta (30) años desde su fundación, veinte (20) de los cuales lleva aliado con el que ahora es su cliente más importante, Inelectra, empresa líder en el área de ingeniería, procura y construcción de proyectos de gran envergadura en los sectores petrolero, petroquímico y de generación eléctrica.

Controlca por tanto, tiene una amplia trayectoria que le ha permitido posicionarse entre los mejores en el ámbito de la consultoría en el campo de las telecomunicaciones.

Si bien, es cierto que Controlca tiene esta trayectoria, requiere mejoras en su sistema de gestión y control, específicamente de las horas – hombre llevadas a cabo por los líderes de proyecto y demás ingenieros involucrados en las tareas de diversos proyectos que la empresa ejecuta.

De allí entonces surge la motivación de desarrollar una solución que permita gestionar y planificar eficientemente la utilización de las horas – hombre para la realización de proyectos, viendo esta problemática desde el punto de vista interno de la empresa, para dar una respuesta más acertada y cercana a la realidad, con miras a conducir el proyecto a una solución posible y aplicable en un lapso mediano de tiempo, para así mejorar este proceso interno de la empresa y el impacto que tiene sobre su principal cliente Inelectra, en términos de eficiencia y de entregar productos de alta calidad a menores costos y en el menor tiempo posible.

Específicamente, la necesidad de la Empresa es conseguir un manejo eficiente de sus proyectos y de los recursos asignados a los mismos, así como la optimización de la asignación de tareas al recurso humano capacitado en base al perfil de conocimiento de los mismos. Para ello debe optimizar los procesos de planificación y gestión de proyectos de ingeniería en telecomunicaciones, haciendo énfasis en la planificación, más que en la gestión.

Controlca, en su afán de encontrar mejoras continuas y con el fin de solventar lo anteriormente expuesto, busca el rediseño de sus procesos internos, con el fin de incorporar nuevos procesos y tecnologías que le permitan una planificación, gestión, administración y seguimiento de todos sus proyectos de telecomunicaciones de forma óptima y eficiente que brinde mejores resultados, tangibles y medibles, tanto para ella como para su principal cliente, Inelectra.

En lo que sigue, este documento representa una síntesis académica que posee el mayor nivel de detalle posible, de cada una de las diferentes etapas a desarrollar, para así llevar a cabo este proyecto y así lograr el éxito deseado.

## **2. ANTECEDENTES DE LA INDUSTRIA Y LA EMPRESA**

### **2.1. La Empresa: Controlca**

CONTROLCA S.A., desde ahora Controlca, es una empresa venezolana, fundada en 1977, por el ingeniero Evaristo Pilo, inicia operaciones en 1978 incorporándose en ese año los ingenieros Eduardo Stiassni y Santiago Rodríguez. Los servicios que ofrecían estaban centrados en servicios profesionales de Ingeniería en consultoría en los campos de telecomunicaciones y sistemas de información.

Dentro de las actividades que ofrece actualmente la empresa se encuentran los siguientes:

- Ingeniería de Telecomunicaciones e Informática
- Sistemas de Administración Telefónica
- Inspecciones de campo
- Estudios de Factibilidad
- Ingeniería Conceptual
- Ingeniería Básica
- Ingeniería de Detalle
- Asistencia Técnica

- Consultoría
- Gerencia de Proyectos
- Inspección y supervisión de la puesta en marcha
- IPGC

Controlca cuenta con una nómina de veinticinco (25) personas que prestan servicio a tiempo completo en la empresa, entre los que se encuentran veinte (20) ingenieros y un (1) técnico , especializados en varios campos técnicos, principalmente: telecomunicaciones, electricidad, electrónica, computación, administración y mercadeo. Cabe destacar que la cantidad de personas que laboran en Controlca puede llegar a ser variable dependiendo de la rotación de personal.

La empresa está en capacidad de cubrir todos los aspectos de un proyecto de ingeniería en las áreas señaladas, incluyendo:

- Ingeniería básica de redes de telecomunicaciones públicas.
- Ingeniería de detalle de redes de telecomunicaciones públicas.
- Ingeniería conceptual de redes de comunicaciones privadas (voz y datos).
- Ingeniería básica de redes de comunicaciones privadas (voz y datos).
- Ingeniería de detalle de redes de comunicaciones privadas (voz y datos).
- Evaluación de sistemas existentes.
- Planificación a corto, mediano y largo plazo.
- Análisis de factibilidad.
- Análisis técnico-económico de alternativas.
- Elaboración de especificaciones para adquisición de equipos, incluyendo las de software en aquellos casos que así lo requieran.
- Análisis técnico-económico de ofertas de suplidores.
- Asistencia en la implementación de un sistema.
- Supervisión y coordinación en la instalación y/o puesta en servicio de equipos.
- Administración, supervisión y/o inspección de proyectos.
- Servicios de mantenimiento y elaboración de programas de mantenimiento.
- Adiestramiento.

- Gerencia de construcción de sistemas de telecomunicaciones

Las áreas de desarrollo de los proyectos están fundamentadas en dos grandes sistemas, Sistemas de Telecomunicaciones y Sistemas de Vigilancia Electrónica, los cuales se fundamentan sobre la infraestructura prevista por los Sistemas de Telecomunicaciones.

- Sistemas de Telecomunicaciones
  - ✓ Cableado estructurado cobre
  - ✓ Cableado fibra óptica
  - ✓ LAN
  - ✓ WAN
  - ✓ Wireless LAN
  - ✓ Telefonía
  - ✓ Videoconferencia
  - ✓ Televisión
  - ✓ Sonorización
  - ✓ Voceo e intercomunicación
  - ✓ Radio comunicaciones
- Sistemas de Vigilancia Electrónica como:
  - ✓ CCTV
  - ✓ Control de acceso
  - ✓ Vigilancia perimetral

Así pues, la empresa tiene la capacidad de cubrir todos los aspectos de un proyecto, desde la concepción básica del sistema hasta su puesta en operación, incluyendo la planificación de las posibles ampliaciones o complementos posteriores.

Desde 1982 Controlca ha desarrollado y comercializado el Sistema de Administración Telefónica SAT en sus versiones SAT 1, SAT 2, SAT 3, SAT IV , SAT 96 y "SAT V plus" versión que es la comercializada actualmente , cuyo objetivo principal es el suministro de información necesaria para administrar los costos telefónicos de las

empresas. El sistema SAT recibe, tarifica y almacena la información de las llamadas telefónicas, cursadas por la central telefónica para su posterior procesamiento.

## **2.2. Alianza estratégica**

Controlca en su intensa búsqueda por mejorar sus servicios, en 1992, se asocia estratégicamente con la mayor empresa de consultoría de Venezuela, Inelectra S.A.C.A., desde ahora Inelectra.

En dicho acuerdo se estableció que el 51 % de las acciones es propiedad de un grupo de profesionales que prestan sus servicios en la empresa y el 49% restante es propiedad de Inelectra S.A.C.A. Sin embargo, las decisiones importantes se toman con el 75% de las acciones.

La dirección operativa de la empresa está a cargo de los ingenieros Josefina Tamayo de Serpa y Eduardo Stiassni, y en la Junta Directiva participan adicionalmente los ingenieros Juan Carlos Gil y María Asunción Ruíz por parte de Inelectra.

Desde su alianza con Inelectra, Controlca, pasa a ser su disciplina de telecomunicaciones incluyéndola dentro de su estructura organizativa.

## **2.3. Misión y Visión de Controlca**

### **2.3.1. Misión**

Controlca, es una empresa privada, cuyo valor fundamental es su recurso humano, afiliada a Inelectra y dedicada a prestar servicios profesionales de ingeniería en el área de telecomunicaciones e informática con la más alta calidad, eficiencia, responsabilidad y ética, con niveles de rentabilidad que aseguren su supervivencia y crecimiento en el mercado; identificándose con las necesidades de sus clientes para satisfacerlos de manera integral.

### **2.3.2. Visión**

Participar activa y eficientemente en el desarrollo del mercado de servicios profesionales de ingeniería en el área de telecomunicaciones e informática, manteniendo el liderazgo sin compromiso comercial con fabricantes de equipo alguno.

### **2.4. Objetivos**

- Cumplir con la misión y visión
- Mantener un sistema de gestión de la calidad basado en la satisfacción del Cliente y partes interesadas
- Minimizar el desperdicio (retrabajo)
- Incrementar el valor de la Empresa a través de sus procesos
- Impartir adiestramiento para alcanzar altos niveles de calidad y personal mejor formado
- Mejora continua de los procesos
- Entender y dar conformidad a los requerimientos del Cliente y partes interesadas y de la Organización.

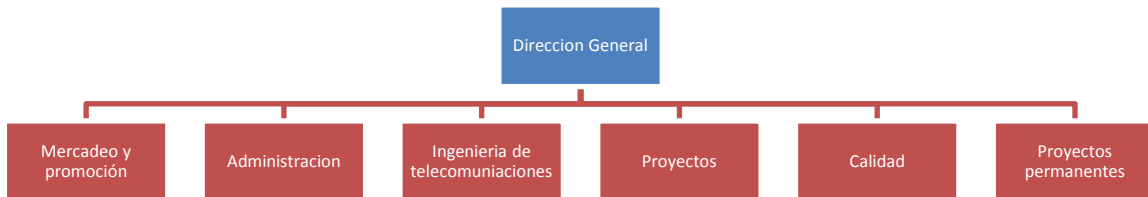
### **2.5. Estructura organizativa**

La organización de Controlca es dinámica adaptándose a los proyectos que se desarrollan en un momento determinado.

A cada proyecto se le asigna un Gerente de Proyecto quien es responsable por éste en cuanto a la calidad del trabajo, tiempo de ejecución, control del costo del proyecto y quien además ejerce la representación de Controlca ante el cliente.

De acuerdo a las exigencias de cada proyecto se asigna el personal idóneo necesario y el apoyo de nuestros servicios internos y, si es necesario, se acude a consultores e ingenieros en telecomunicaciones externos para cumplir con el compromiso contraído.

Actualmente la organización de Controlca se estructura según el siguiente organigrama, en el cual se destacan las siguientes unidades organizativas:



- Mercadeo y promoción: encargado de la elaboración de ofertas y, conjuntamente con la dirección de la empresa, se ocupa de la atención a los clientes.
- Administración: se encarga de la gestión del personal, de la contabilidad, de la facturación y de la logística de la oficina.
- Ingeniería de telecomunicaciones: encargada de la elaboración de proyectos.
- Proyectos: encargada del control y seguimiento de la ejecución de los proyectos.
- Calidad: encargada del sistema de calidad de la empresa.
- Proyectos permanentes: desarrollo, instalación y mantenimiento de los sistemas de administración telefónica SAT. Servicio de mantenimiento y apoyo en la operación del sistema telefónico de Inelectra.

## 2.6. Clientes

En el siguiente listado se muestran las empresas de los diferentes sectores del país que han contratado los servicios de Controlca para asesorías y para la elaboración de proyectos de telecomunicaciones:

- Sector bancario
- Sector empresarial
- Sector consultoría
- Sector industrias básicas
- Sector comunicaciones
- Sector petrolero



- Sector educativo
- Sector clínicas

## **2.7. INELECTRA S.A.C.A.**

INELECTRA S.A.C.A., desde ahora Inelectra, fundada en el año 1968, es una empresa de capital venezolano dedicada a la oferta de servicios de ingeniería y construcción para las industria petrolera, petroquímica, acero, aluminio, electricidad, telecomunicaciones y transporte masivo, que ejecuta proyectos de gran envergadura en Venezuela y el exterior. Adicionalmente, participa como socio propietario en inversiones relacionadas a sus principales áreas de negocio: Ingeniería y Construcción, así como Servicios de Petróleo.

### Servicios profesionales de ingeniería y construcción

- Estudio de factibilidad
- Ingeniería conceptual y básica
- Ingeniería de detalle
- Gerencia de proyectos
- Construcción
- Gerencia de construcción
- Ingeniería de procesos
- Procura
- Gerencia de procura
- Parada de plantas
- Arranque y puesta en marcha

Los segmentos de mercado que atiende Inelectra están dirigidos hacia las siguientes áreas que representan básicamente al área de la energía:

- Facilidades de producción de crudos y gas
- Producción y procesamiento de gas
- Refinación y mejoramiento de crudos

- Petroquímica
- Transporte, distribución y almacenamiento de hidrocarburos
- Instalación de plataformas costafuera
- Generación y transmisión de electricidad

Es importante señalar que, para el desarrollo del presente trabajo, se considera que Controlca posee un Cliente del Proyecto de Tesis, el cual, es Inelectra. De esta forma, el Cliente Final es el Cliente Inelectra.

### **3. MARCO TEORICO CONCEPTUAL**

En este capítulo se detalla la base teórica conceptual sobre la cual se apoya el proyecto. Específicamente este capítulo incluye los conceptos necesarios para la comprensión y aplicación del proceso de asignación optimizada de tareas a personal calificado, basándonos en la teórica de programación de proyectos con recursos restringidos RCPSP, *Resource-Constrained Project Scheduling Problem*, la cual permitirá entender teóricamente como se obtendrán los resultados y objetivos esperados. Los problemas de programación de proyectos están constituidos por actividades, recursos, relaciones de precedencia y función objetivo.

Así mismo, dado que la asignación optimizada de actividades busca realizar mejoras al control y gestión de proyectos de consultoría en ingeniería, es que se ha estudiado en detalle qué hay detrás de esta ciencia, optando entonces, por la utilización del estándar de facto que existe en la industria, como el publicado por el PMI (Project Management Institute) a través del PMBoK (Project Management Book of Knowledge).

#### **3.1. RCPSP: Resource-Constrained Project Scheduling Problem**

Un proyecto consiste de un número de actividades, también conocidas como trabajos, operaciones o tareas. Para completar el proyecto exitosamente, cada actividad puede ser procesada en uno de varios posibles modos. Cada modo representa una manera diferente de realizar la actividad que está siendo considerada. El modo

determina la duración de la actividad, los requerimientos de los diferentes tipos de recursos y cualquier otra característica asociada con la actividad.

La programación de proyectos con recursos limitados es un ambiente en el cual se deben procesar un conjunto de actividades sujetas a restricciones de precedencias y recursos, siendo éstos últimos compartidos por varias actividades. Por lo tanto el problema radica en realizar la asignación de recursos optimizando alguna función objetivo. Esto supone más que un problema únicamente de tipo cualitativo, ya que el tema da origen a definiciones y formalizaciones de tipo matemático cuando se trata de resolver el problema general de programación de tareas.

Gran parte de la investigación realizada en el área de la programación de proyectos con recursos limitados se concentra en minimizar la duración del proyecto en condiciones determinísticas, mediante la asignación optimizada de actividades o tareas. Las actividades están sujetas a relaciones de precedencia inicio-fin sin holguras. Puede ser el caso en que las actividades tienen un único modo de realizarse y todos los recursos son renovables o el caso más general de múltiples modos de ejecución y recursos renovables y no renovables.

El problema de programación de proyectos con recursos restringidos y un único modo de ejecución de las actividades (RCPSP) es una generalización del problema de Job Shop en el cual las tareas pueden usar múltiples recursos y los recursos pueden tener una capacidad mayor a uno.

El problema del RCPSP, y en general cualquier problema de *scheduling*, es un problema de optimización combinatoria. La función del *scheduling* es la asignación de recursos limitados a tareas a lo largo del tiempo y tiene como finalidad la optimización de uno o más objetivos.

Los recursos pueden ser máquinas en un taller, pistas en un aeropuerto, ladrillos en una construcción, unidades de procesamiento en un ambiente computacional, personal capacitado, etc. Como tareas se pueden tener operaciones de un proceso de producción, despegues y aterrizajes en un aeropuerto, tareas de un proyecto,

ejecuciones de un programa computacional, etc. Cada tarea puede tener diferentes niveles de prioridad, así como tiempos de posibles inicios. Los objetivos pueden tomar varias formas: minimizar los tiempos de finalización de la última tarea, minimizar el número de tareas luego de una fecha de entrega acordada, etc.

El RCPSP puede ser un problema difícil tanto desde el punto de vista técnico como de implementación. El tipo de dificultades encontradas en los aspectos técnicos son similares a las encontradas en otras ramas de optimización combinatoria y modelado estocástico. Las dificultades encontradas desde el punto de vista de la implementación son de distintas clases y están relacionadas al modelado de problemas de *scheduling* del mundo real y la recuperación de información.

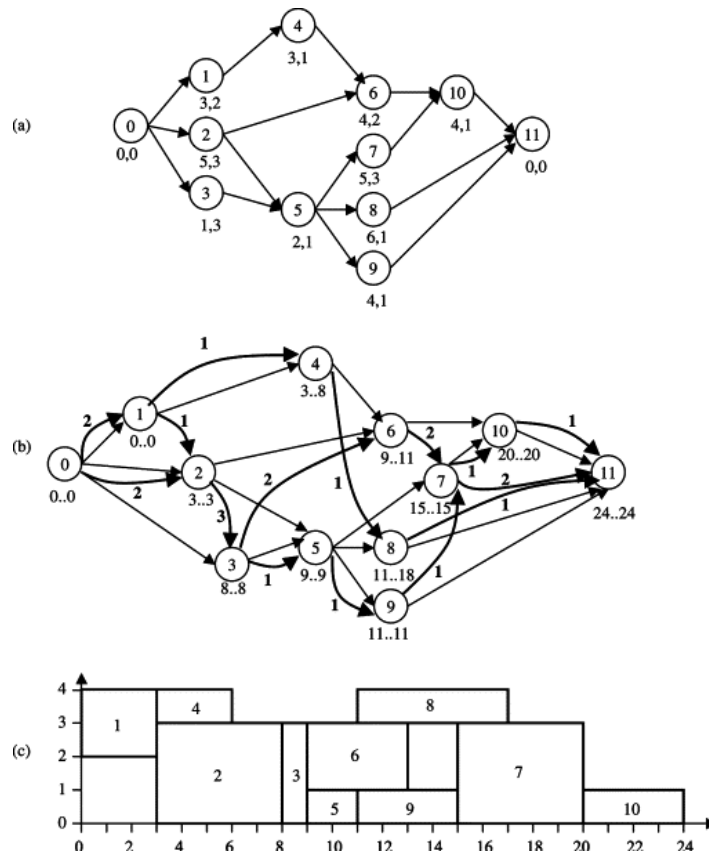
### 3.1.1. Definición del problema

El problema del RCPSP, corresponde a un tipo de problema de planificación de tareas, el cual a su vez, es un problema de optimización con ciertas características; en donde uno de los objetivos principales es la óptima asignación de recursos.

Consiste básicamente en planificar un conjunto de  $N$  trabajos  $\{J_1, \dots, J_N\}$  sobre un conjunto de  $M$  recursos  $\{R_1, \dots, R_M\}$ . Cada trabajo  $J_i$  consta de un conjunto de operaciones o tareas  $\{\theta_{i1}, \dots, \theta_{iN}\}$  que deben ser ejecutadas de forma secuencial. Cada tarea  $\theta_{ij}$  tiene asociado un tiempo de procesamiento sin interrupción de  $du_{ij}$  unidades de tiempo durante el cual requiere del uso exclusivo de un único recurso. Cada trabajo tiene un tiempo de inicio más temprano y en ocasiones se considera también un tiempo de término más tardío, lo que obliga a que los tiempos de inicio de las tareas tomen valores en dominios finitos. Cabe considerar que las operaciones de un mismo trabajo deben ejecutarse en un orden determinado, de forma que  $\theta_{(i+1)j}$  no puede comenzar hasta que  $\theta_{ij}$  haya terminado completamente. Además, las operaciones que comparten un mismo recurso son no interrumpibles y mutuamente exclusivas.

El objetivo consiste en encontrar una planificación factible, es decir, una asignación de tiempos de inicio  $st_{ij}$  para cada una de las tareas, que minimice el *makespan* ( $C_{max}$ ) o tiempo de finalización de la última tarea. El RCPSP se representa a

través de un grafo conocido como grafo disyuntivo, como el que se muestra en la siguiente figura.



Una formulación matemática para el RCPSP es la siguiente:

$$\text{Min } z_p = \sum_{t=es_n}^{ls_n} t \times \varepsilon_{nt} \quad (1)$$

Sujeto a:

$$\sum_{t=es_j}^{ls_j} \varepsilon_{jt} = 1; j \in J \quad (2)$$

$$\sum_{t=es_i}^{ls_i} t \times \varepsilon_{it} - \sum_{t=es_j}^{ls_j} t \times \varepsilon_{jt} \geq d_j; (i, j) \in H \quad (3)$$

$$\sum_{j \in J} \sum_{r=\sigma(t,j)}^t r_{jk} \times \varepsilon_{jr} \leq R_k; t=0, \dots, T_{\max}; K=1, \dots, m \quad (4)$$

$$\varepsilon_{jt} \in \{0,1\}; j \in J; t=es_j, \dots, ls_j \quad (5)$$

La ecuación (1) representa la función objetivo de cada modelo respectivamente, *makespan* o duración del proyecto. La ecuación (2) representa el tiempo de inicio de cada actividad en el modelo de programación binaria, el cual debe ser único dada la característica de no interrupción de las actividades, es decir, las que obligan a que una actividad, una vez iniciada, debe continuarse hasta su terminación. La ecuación (3) representa las restricciones de precedencia en cada modelo respectivamente, una actividad sólo puede iniciar una vez terminadas todas las actividades predecesoras. Las ecuación (4) representan las restricciones de recursos de cada modelo respectivamente; en cualquier tiempo la cantidad de recursos utilizados por todas las actividades en ejecución no debe superar la disponibilidad de cada tipo de recurso correspondiente. La ecuación (5) indica que las variables de decisión de la formulación binaria únicamente pueden tomar los valores 0 ó 1.

### **3.1.2. Metodología**

Para solucionar el RCPSP se han utilizado diversos enfoques, los cuales se pueden clasificar como métodos exactos y métodos heurísticos.

#### **3.1.2.1. Métodos exactos**

La principal característica de estos algoritmos es que obtienen una solución óptima al problema, cuando ésta existe.

Sin embargo, la naturaleza combinatoria de estos problemas y su pertenencia a la clase de problemas *NP-Hard*, hace que sean prácticamente imposibles de resolver en tiempos razonables, aún mediante el uso de los computadores más potentes existentes hoy en día. La experiencia de otros autores indica que sólo es posible obtener la solución mediante métodos exactos para instancias del problema con menos de 60

actividades<sup>1</sup>. Sin embargo, un proyecto de 60 actividades se puede considerar pequeño si se compara con los casos reales; por lo tanto, toda discusión acerca de la solución del RCPSP se centra en algoritmos aproximados que cada autor, desde su enfoque, plantea como eficientes.

### **3.1.2.2. Métodos heurísticos**

En contraposición a los métodos exactos, los heurísticos no garantizan la obtención de soluciones óptimas. Sin embargo, suelen obtener soluciones satisfactorias en tiempos de procesamiento generalmente polinomiales.

Los métodos heurísticos más usados son Búsqueda Tabú, Recocido Simulado, Búsqueda en Vecindarios Variables, Algoritmos Genéticos, Optimización de Colonias de Hormigas, Ramificación y poda (*Branch & Bound*), entre otros.

En este proyecto se utilizará el método *Branch & Bound* mediante la herramienta RESCON, la cual utiliza métodos heurísticos para resolver problemas de planificación de recursos limitados.

### **3.1.2.3. Ramificación y poda (*Branch & Bound*)**

Las técnicas de ramificación y poda hacen uso de una estructura de árbol construida dinámicamente, como forma de representar el espacio de todas las secuencias posibles. La búsqueda comienza en el nodo raíz y se continúa hasta llegar a un nodo hoja. Cada nodo en un nivel  $p$  de la búsqueda representa una secuencia parcial de  $p$  operaciones.

El procedimiento de poda selecciona la operación con la que continuará la búsqueda y se basa en una estimación de una cota inferior y la mejor cota superior

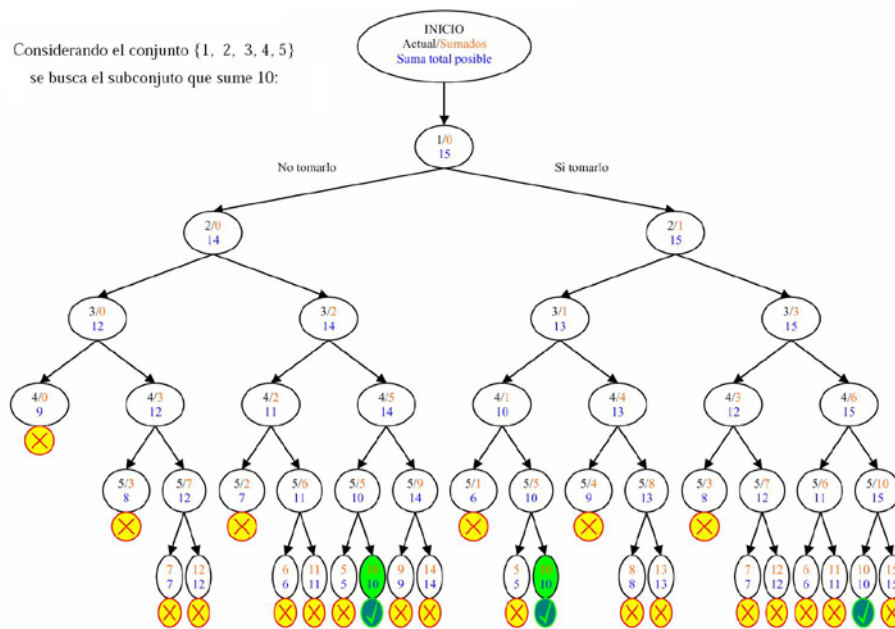
---

<sup>1</sup> V. Valls, F. Ballestín, S. Quintanilla. "Justification and RCPSP: a technique that pays". European Journal of Operational Research.

alcanzada hasta el momento. En los métodos de ramificación y poda es esencial afinar las cotas, ya que eso previene la necesidad de buscar en secciones grandes del espacio de soluciones.

La técnica de Ramificación y poda se suele interpretar como un árbol de soluciones, donde cada rama nos lleva a una posible solución posterior a la actual. La característica de esta técnica es que el algoritmo se encarga de detectar en qué ramificación las soluciones dadas ya no están siendo óptimas, para “podar” esa rama del árbol y no continuar malgastando recursos y procesos en los casos los que se alejan de la solución óptima.

A continuación se presenta un ejemplo de árbol creado bajo esta técnica.



Una simple interpretación del algoritmo en pseudocódigo sería esta:



```

Funcion RyP {
  P = Hijos(x,k)
  mientras ( no vacio(P) )
  x(k) = extraer(P)
  si esFactible(x,k) y G(x,k) < optimo
  si esSolucion(x)
  Almacenar(x)
  sino
  RyP(x,k+1)
}

```

### 3.1.3. RESCON

El RESCON es un software desarrollado para resolver problemas tipo RCPSP, *Resource-Constrained Project Scheduling Problem*. La herramienta contempla las siguientes características y funcionalidades.

- Representación de las actividades en forma de grafo
- Procedimientos de programación heurística (50 en total)
- Múltiples perfiles de recursos
- Generación de cartas de Gantt
- Análisis comparativo de los algoritmos
- Métricas del proyecto

Este software será el empleado para modelar algunos proyectos y demostrar, con una prueba de concepto, que la aplicación de éste genera resultados eficientes en cuanto a la asignación de recursos a proyectos de telecomunicaciones.

### 3.2. Marco Teórico Conceptual: PMBoK

Como se explica en la introducción PMI: PMBoK es un documento que incluye las mejores prácticas relacionadas al control y gestión de proyectos, recopiladas del éxito de muchas empresas, tanto así que se ha convertido en un estándar avalado por la ANSI. En este proyecto se utilizarán estos conceptos y serán aplicados al diseño de

procesos atingente con este trabajo, por lo que es necesario su conocimiento, el cual se detalla a continuación.

El Project Management Institute (PMI) creó en 1996 la primera edición de “A Guide to the Project Management Body of Knowledge” llevando al mundo un documento que, poco a poco, fue calando en las industrias y en la administración de proyectos, convirtiéndose en un estándar avalado por la ANSI en el año 2000. Actualmente, en su cuarta edición de 2008, el PMBoK es el documento de referencia obligado para cualquier persona que desee mejorar su gerencia de proyectos, certificarse o simplemente incrementar el éxito de sus proyectos; y para cualquier organización que desee implementar procesos y metodologías eficaces para lograr el éxito de sus proyectos.

El PMBoK compite con otros modelos de gestión de proyectos como el de la *Association for Project Management (APM)* y *Prince* (en Reino Unido); sin embargo, se ha posicionado a nivel mundial como un estándar en gerencia de proyectos y las certificaciones otorgadas sobre éste, como *Certificate Associate in Project Management (CAPM)* y *Project Management Professional (PMP)* son las más reconocidas por las empresas y las más buscadas por los practicantes; es por esto que se utilizarán como marco teórico-conceptual de este proyecto.

El PMBoK es un compendio de mejores prácticas, heredadas de diversas industrias y disciplinas que conforman un modelo metodológico. El PMBoK en sí no es una metodología que “deba” ser seguida al pie de la letra; de hecho, el mismo documento indica que los procesos y sus relaciones deben ser personalizados a las necesidades del proyecto y de la empresa. El PMBoK es sólo una guía, muy completa y elaborada, de lo que normalmente un gerente de proyectos debe llevar a cabo, explicado en un buen nivel de detalle y separando procesos que normalmente se llevan a cabo de forma simultánea.

### 3.2.1. Definiciones

PMBoK establece varias definiciones en función de entender un lenguaje común, las cuales serán sintetizadas a continuación, en el presente capítulo.

**Proyecto:** según PMBoK un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único, definiendo un inicio y un final bien determinado, ya sea cuando se logran los objetivos del proyecto o cuando se decide dar fin al mismo sin lograrlos.

**Gestión de proyectos:** la gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del mismo. A menudo los proyectos se utilizan como medio para cumplir con el plan estratégico de una organización.

**Oficina de dirección de proyectos o PMO:** una oficina de dirección de proyectos es un cuerpo o entidad dentro de una organización que tiene varias responsabilidades asignadas con relación a la dirección centralizada y coordinada de los proyectos. Las responsabilidades pueden abarcar desde proveer funciones de apoyo para la gestión hasta la responsabilidad de dirigir algún proyecto directamente. Puede delegarse la autoridad necesaria para actuar como un interesado integral y tomar decisiones claves al comienzo de cada proyecto, para hacer sugerencias o para terminar proyectos o tomar otras medidas (según se requiera) a fin de mantener la coherencia con los objetivos del negocio. Una función fundamental de esta oficina es brindar apoyo a los jefes de proyecto de diferentes formas, entre ellas:

- Gestionar recursos compartidos por todos los proyectos.
- Identificar y desarrollar una metodología, mejores prácticas y normas para la dirección de proyectos.
- Instruir, orientar, capacitar y supervisar.
- Vigilar el cumplimiento de las normas, procedimientos y plantillas mediante auditorías.

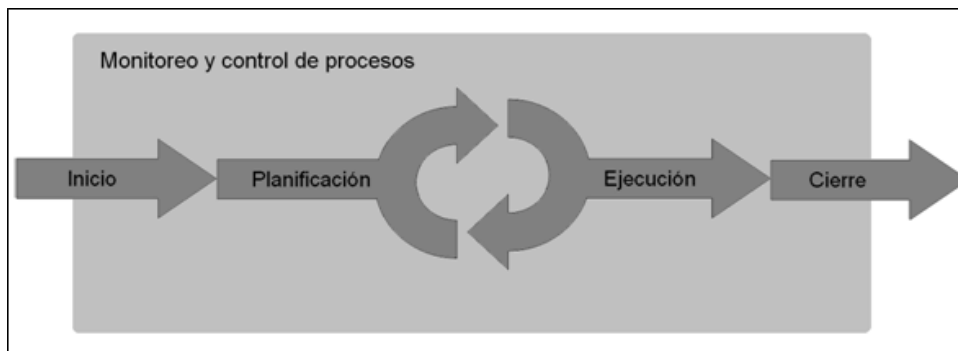
- Desarrollar y gestionar políticas, procedimientos, plantillas y otra documentación compartida del proyecto.
- Coordinar la comunicación entre proyectos.

**Director o jefe del proyecto:** Es la persona encargada de alcanzar los objetivos del proyecto, quien debe contar con las siguientes características:

- Conocimiento: se refiere a cuánto sabe sobre gestión de proyectos.
- Desempeño: se refiere a cuánto puede lograr si aplica los conocimientos que posee.
- Personal: tiene relación con la manera en que el director se comporta cuando ejecuta el proyecto, abarcando actitudes, personalidad y liderazgo, entendido como la capacidad de guiar al equipo mientras se cumplen objetivos y se equilibran las restricciones del mismo.

### 3.2.2. Ciclo de vida de un proyecto

Los proyectos varían en tamaño y complejidad. Todos los proyectos, sin importar cuán pequeños, grandes, sencillos o complejos sean, pueden configurarse dentro de la estructura del ciclo de vida que se muestra en la siguiente figura: Ejemplo de un proyecto de una sola fase.



Si bien existen varias posibilidades para trabajar proyectos con varias fases, sean esta secuenciales, iterativas o superpuestas, por ejemplo, se debe mantener la

idea de gobernabilidad del proyecto a lo largo de su ciclo de vida, que no es más que proporcionar un método integral y coherente de controlar el proyecto y asegurar el éxito.

### 3.2.3. Interesados

Los interesados (también conocidos como *stakeholders*) son personas u organizaciones que participan activamente en el proyecto y cuyos intereses pueden verse afectados positiva o negativamente por la ejecución o término del proyecto. Los interesados también pueden ejercer influencia sobre el proyecto, los entregables y los miembros del equipo. Se debe identificar a estos actores con el fin de determinar los requisitos y las expectativas de todas las partes involucradas. Las definiciones sobre cada interesado pueden ser consultados en detalle en la guía del PMBoK, mientras acá se dará una breve descripción y su posicionamiento, se puede ver en la siguiente figura: Interesados en el proyecto.



Los interesados, mostrados en la figura, se definen a continuación:

- **Clientes / Usuarios:** personas u organizaciones que usarán el producto.
- **Patrocinador:** persona o grupo que proporciona los medios financieros para el proyecto.
- **Administrador de portafolio:** responsables de la gobernabilidad de alto nivel de un programa o conjunto de proyectos.
- **Administrador de programa:** responsables de la gestión coordinada de proyectos relacionados.
- **PMO:** cuerpo o entidad que tiene varias responsabilidades relacionadas con la dirección centralizada y coordinada de los proyectos.
- **Jefe de proyecto:** encargado de alcanzar los objetivos del proyecto.
- **Equipo del proyecto:** equipo conformado por el director, equipo de dirección y otros miembros.
- **Directores funcionales:** personas clave que desempeñan el rol de gestores dentro de un área administrativa o funcional de una empresa.
- **Gerente operaciones:** aquellos que desempeñan una función de gestión en un área medular de una empresa.
- **Vendedores del negocio:** compañías externas que celebran un contrato para el proyecto.

#### **3.2.4. Descripción de los procesos por etapa**

A continuación se describe cada proceso indicando con un nivel de detalle las actividades que están relacionadas según la fase o etapa en la que se encuentra el proyecto.

##### **3.2.4.1. Proceso de Iniciación**

Son aquellos procesos realizados para definir un nuevo proyecto o una nueva fase de un proyecto ya existente, mediante la obtención de la autorización para comenzar dicho proyecto o fase.

Las actividades que considera son:

1. **Desarrollar el Acta de Constitución del Proyecto:** documento formal que autorice un proyecto.
2. **Identificar a los Interesados:** identificar a todas las personas impactadas por el proyecto.

#### 3.2.4.2. Proceso de Planificación

Aquellos procesos requeridos para establecer el alcance del proyecto, refinar los objetivos y definir el curso de acción necesario para alcanzar los objetivos para cuyo logro se emprendió el proyecto.

Las actividades que considera son:

1. **Desarrollar el Plan para la Dirección del Proyecto:** documentar las acciones necesarias para definir, preparar, integrar y coordinar todos los planes subsidiarios.
2. **Recopilar Requisitos:** definir y documentar las necesidades de los interesados para cumplir con los objetivos del proyecto.
3. **Definir el Alcance:** descripción detallada del proyecto.
4. **Crear la EDT (Estructura de Desglose del Trabajo):** subdividir los entregables y el trabajo en componentes más pequeños y más fáciles de dirigir.
5. **Definir las Actividades:** identificar acciones específicas a ser realizadas para elaborar los entregables del proyecto.
6. **Secuenciar las Actividades:** identificar y documentar las relaciones entre las actividades del proyecto.
7. **Estimar los Recursos de las Actividades:** estimar tipo y cantidad de personas y/o suministros requeridos.
8. **Estimar la Duración de las Actividades:** estimar aproximadamente la cantidad de períodos de trabajo necesarios para finalizar cada actividad con los recursos estimados.
9. **Desarrollar el Cronograma:** analizar el orden de actividades, duración, requisitos y restricciones.
10. **Estimar Costos:** desarrollar una aproximación de los recursos monetarios necesarios.

11. **Determinar el Presupuesto:** sumar los costos estimados para establecer una línea base de costos.
12. **Planificar la Calidad:** identificar requisitos de calidad y/o normas para el proyecto, documentando cómo se demostrará el cumplimiento con los mismos.
13. **Desarrollar el Plan de Recursos Humanos:** identificar roles dentro de un proyecto, responsabilidades, habilidades y comunicación en la dirección del personal.
14. **Planificar las Comunicaciones:** determinar necesidades de información de los interesados en el proyecto para definir cómo abordar las comunicaciones.
15. **Planificar la Gestión de Riesgos:** definir cómo realizar las actividades de gestión de riesgos.
16. **Identificar Riesgos:** determinar riesgos que puedan afectar el proyecto y documentarlos.
17. **Realizar Análisis Cualitativo de Riesgos:** priorizar los riesgos para realizar otros análisis, evaluando y combinando la probabilidad de ocurrencia e impacto de dichos riesgos.
18. **Realizar Análisis Cuantitativo de Riesgos:** analizar numéricamente el efecto de los riesgos identificados sobre los objetivos generales del proyecto.
19. **Planificar la Respuesta a los Riesgos:** desarrollar opciones y acciones para mejorar las oportunidades y reducir las amenazas a los objetivos del proyecto.
20. **Planificar las Adquisiciones:** documentar las decisiones de compra para el proyecto.

#### **3.2.4.3. Proceso de Ejecución**

Aquellos procesos realizados para completar el trabajo definido en el plan para la dirección del proyecto a fin de cumplir con las especificaciones del mismo.

1. **Dirigir y Gestionar la Ejecución del Proyecto:** ejecutar el trabajo definido en el plan.



2. **Realizar Aseguramiento de Calidad:** auditar los requisitos de calidad y los resultados obtenidos a partir de medidas de control de calidad.
3. **Adquirir el Equipo del Proyecto:** confirmar los recursos humanos disponibles y formar el equipo necesario para completar las asignaciones del proyecto.
4. **Desarrollar el Equipo del Proyecto:** mejorar las competencias, la interacción de los miembros del equipo y el ambiente para lograr un mejor desempeño.
5. **Dirigir el Equipo del Proyecto:** dar seguimiento al desempeño de los miembros del equipo, proporcionar retroalimentación, resolver problemas y gestionar cambios.
6. **Distribuir la Información:** poner la información relevante a disposición de los interesados.
7. **Gestionar las Expectativas de los Interesados:** comunicarse y trabajar en conjunto con los interesados para satisfacer sus necesidades y abordar los problemas.
8. **Efectuar Adquisiciones:** obtener respuestas de los vendedores, seleccionar uno y adjudicar un contrato.

#### **3.2.4.4. Proceso de Seguimiento y Control**

Aquellos procesos requeridos para dar seguimiento, analizar y regular el progreso y el desempeño del proyecto, para identificar áreas en las que el plan requiera cambios y para iniciar los cambios correspondientes.

1. **Dar Seguimiento y Controlar el Trabajo del Proyecto:** revisar, analizar y regular el avance a fin de cumplir con los objetivos de desempeño definidos.
2. **Realizar Control Integrado de Cambios:** revisar todas las solicitudes de cambios, aprobarlos y gestionar los mismos.
3. **Verificar el Alcance:** formalizar la aceptación de los entregables del proyecto.
4. **Controlar el Alcance:** dar seguimiento al estado de alcance del proyecto y gestionar cambios a la línea base del alcance.

5. **Controlar el Cronograma:** dar seguimiento a la situación del proyecto para actualizar el avance del mismo y gestionar cambios a la línea base del cronograma.
6. **Controlar Costos:** dar seguimiento a la situación del proyecto para actualizar el presupuesto del mismo y gestionar cambios a la línea base de costo.
7. **Realizar Control de Calidad:** dar seguimiento y registrar los resultados de la ejecución de actividades de control de calidad, a fin de evaluar el desempeño y realizar los cambios necesarios.
8. **Informar el Desempeño:** recopilar y distribuir información sobre el desempeño, como informes de estado, mediciones de avance y proyecciones.
9. **Dar Seguimiento y Controlar los Riesgos:** implementar planes de respuesta a los riesgos identificados, residuales y nuevos riesgos y evaluar la efectividad del proceso.
10. **Administrar las Adquisiciones:** gestionar las relaciones de adquisiciones, supervisar el desempeño del contrato y efectuar cambios y correcciones según sea necesario.

#### **3.2.4.5. Proceso de Cierre**

Aquellos procesos realizados para finalizar todas las actividades a través de todos los grupos de procesos, a fin de cerrar formalmente el proyecto o una fase del mismo.

1. **Cerrar el Proyecto o Fase:** finalizar todas las actividades a través de todos los grupos de procesos de dirección de proyectos para completar el formalmente proyecto.
2. **Cerrar las Adquisiciones:** finalizar cada adquisición del proyecto.

### 3.2.5. Agrupamiento de Procesos por Área de Gestión

Los mismos procesos enunciados anteriormente, agrupados por etapa, pueden ser vistos según el área de gestión que afecten, según se describe a continuación y se puede ver en la Figura 6: Agrupamiento de Procesos por Área de Gestión.

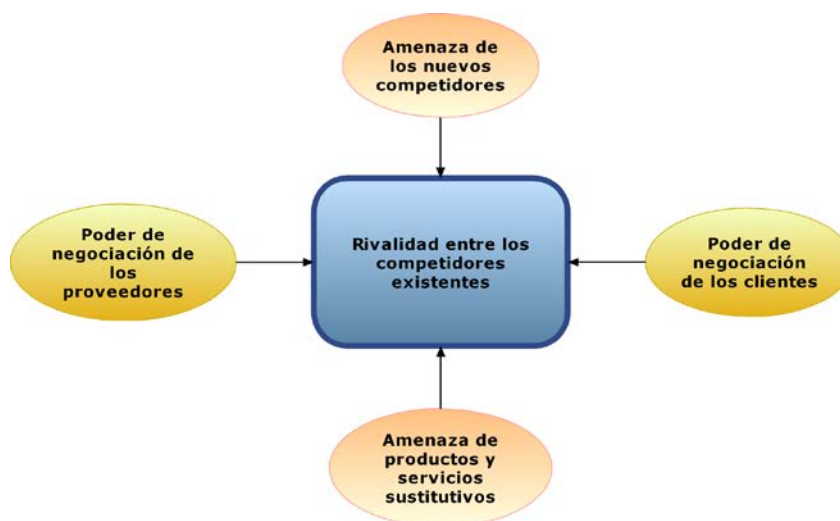
- **Gestión de Integración:** Procesos requeridos para integrar todas las actividades, documentos y recursos del proyecto.
- **Gestión de Alcance:** Procesos requeridos para identificar todo el trabajo requerido (ni más ni menos) para obtener los entregables del proyecto y cumplir los objetivos.
- **Gestión de Tiempo:** Procesos requeridos para asegurar que el proyecto es finalizado a tiempo.
- **Gestión de Costos:** Procesos requeridos para asegurar que el proyecto es finalizado dentro de un presupuesto aprobado.
- **Gestión de Calidad:** Procesos requeridos para asegurar que el proyecto cumple los requerimientos y necesidades por los cuales fue emprendido.
- **Gestión de Comunicaciones:** Procesos requeridos para asegurar la generación, distribución, almacenamiento y disposición última de toda la información del proyecto, a tiempo y de forma adecuada.
- **Gestión de Recursos Humanos:** Procesos requeridos para administrar eficientemente la gente que participa en el proyecto.
- **Gestión de Riesgos:** Procesos requeridos para identificar, analizar y responder efectivamente a los riesgos del proyecto.
- **Gestión de Adquisiciones:** Procesos requeridos para adquirir bienes y servicios fuera de la organización del proyecto.

## 4. ANÁLISIS DEL MEDIO

Para un mejor entendimiento de la dinámica de la industria y de la posición de la empresa frente a ella, se presentará un mapa estratégico al estilo del análisis de las Fuerzas de Porter, así como un análisis FODA.

## 4.1. Análisis de las Fuerzas de Porter

Las 5 Fuerzas de Porter es un modelo holístico que propone una guía de reflexión estratégica sistemática que permite analizar cualquier industria en términos de rentabilidad. Fue desarrollado por Michael Porter en 1979 y, según el mismo, la rivalidad con los competidores viene dada por cuatro elementos o fuerzas que, combinadas, crean una quinta fuerza: la rivalidad entre los competidores [2]. Sus interacciones se pueden ver en siguiente figura: “Modelo de Fuerzas de Porter”. A continuación se desarrolla un análisis del medio utilizando este modelo.



### 4.1.1. Poder de negociación de los Compradores o Clientes

1. **Concentración de compradores respecto a la concentración de compañías:** alto, dado que Controlca es el único proveedor de servicios de Telecomunicaciones para Inelectra, tanto para proyectos nacionales como internacionales.
2. **Grado de dependencia de los canales de distribución:** no existe.
3. **Posibilidad de negociación,** especialmente en industrias con muchos costes fijos.
4. **Volumen comprador:** de los varios clientes que posee Controlca para sus diferentes productos, Inelectra representa el 92% de sus ingresos.

5. **Costes o facilidades del cliente de cambiar de empresa:** los costos de cambiar de empresa por parte del cliente, Inelectra, son altos dada la relación de confianza y de conocimiento que ha construido con Controlca desde hace mas de 15 años.
6. **Disponibilidad de información para el comprador:** el cliente tiene acceso a toda la información de la empresa, tanto de las soluciones que ofrece como de los aspectos financieros, sobre los cuales puede ejercer control.
7. **Capacidad de integrarse hacia atrás:** actualmente Controlca es considerada parte de la estructura organizativa de Inelectra.
8. **Existencia de productos sustitutos:** existen otras empresas consultoras que se manejan en el mismo rubro.
9. **Sensibilidad del comprador al precio:** bajo, dado que la alianza estratégica entre ambas empresas hace que la competencia no pueda “igualar” los precios.
10. **Ventaja diferencial (exclusividad) del producto:** dado que la empresa entrega un servicio integral, pasa a ser una ventaja para el cliente contar con un servicio completo de planificación, control y gestión de proyectos de telecomunicaciones.

#### **4.1.2. Poder de negociación de los Proveedores o Vendedores**

Dado que el servicio se provee con recursos internos, no se utilizan proveedores, por lo que no existe el concepto de poder de negociación con ellos.

#### **4.1.3. Amenaza de nuevos entrantes**

1. **Existencia de barreras de entrada:** alta, dado el nivel de integración actual que Controlca tiene con Inelectra.
2. **Economías de escala:** Economías de escala: no existen economías de escala en términos de que mientras más clientes tenga la empresa menores serán los costos pues los desarrollos son a medida.

3. **Valor de la marca:** alto, dado que el trabajo realizado por Controlca es diseñado a la medida del cliente final cumpliendo los altos estándares de calidad de Inelectra.
4. **Costes de cambio:** alto, dada la relación de confianza y de conocimiento que ha construido con Controlca desde hace más de 15 años con Inelectra.
5. **Requerimientos de capital:** alto, dada las exigencias actuales que impone el cliente a sus aliados comerciales.
6. **Ventajas absolutas en coste:** alta, dado que Controlca obtiene márgenes de ganancia menores que la competencia dada la alianza estratégica que mantiene con Inelectra, lo cual es muy difícil de adoptar por los competidores.
7. **Ventajas en la curva de aprendizaje:** Controlca tiene más de 15 años dedicado a proyectos de ingeniería especializados en telecomunicaciones en conjunto con Inelectra, siguiendo sus pautas y estándares de calidad. Además adaptándose a sus requerimientos y exigencias.
8. **Represalias esperadas:** bajo, dado que no existen competidores sustitutos que puedan ejercer las funciones que hoy en día cumple Controlca como aliado estratégico en el rubro de las telecomunicaciones con Inelectra.
9. **Mejoras en la tecnología:** bajo, dado que Inelectra es quien marca la pauta en los temas tecnológicos, a los que debe adaptarse Controlca.

#### 4.1.4. Amenaza de productos sustitutos

1. **Propensión del comprador a sustituir:** baja, dada la relación estratégica y de confianza que ambas empresas comparten.
2. **Precios relativos de los productos sustitutos:** alto, dado que las tarifas por hora hombre acordadas entre ambas empresas es inferior a la que los competidores podrían ofrecer.
3. **Coste o facilidad de cambio del comprador:** bajo, dado que Controlca conoce el “know how” de Inelectra, en cuanto al desarrollo de proyectos en el ámbito de las telecomunicaciones.
4. **Nivel percibido de diferenciación de producto:** alto, dado que el producto entregado se ajusta a las necesidades y estándares de calidad exigidos por Inelectra.

5. **Disponibilidad de sustitutos cercanos:** bajo, ya que se desconoce el know how de la empresa, además de que no se podría competir con la tarifa de horas – hombre que le ofrece Controlca a Inectra, su cliente principal.

#### **4.1.5. Rivalidad entre los competidores**

En términos del desarrollo de productos de telecomunicaciones a la medida del cliente final según las especificaciones y estándares de Inectra, la rivalidad entre los competidores se podría decir que es baja según lo explicado en los puntos anteriores: las barreras en la entrada al mercado son bajas y la propensión del comprador a sustituir es baja.

### **4.2. Análisis FODA**

#### **4.2.1. Fortalezas**

- Información de proyectos desarrollados previamente como base para el desarrollo de otros proyectos.
- Recursos de calidad y con experiencia en el área de las telecomunicaciones.
- Trayectoria reconocida en el área de las telecomunicaciones.
- Capacidad financiera para aplicar cambios en la gestión de procesos
- Existe un conocimiento general de la existencia de la falla en la gestión y control de horas – hombre.
- Existe una decisión corporativa de responder positivamente a la propuesta de mejoramiento de los procesos de gestión de horas – hombre

#### **4.2.2. Oportunidades**

- Existe en Chile una brecha importante de acceso a redes de telecomunicaciones que debe ser resuelta.
- Acertado momento económico del país para invertir en proyectos de Telecomunicaciones.

- Existe una gran cantidad de empresas que reconocen en la gestión, una de sus principales debilidades, por lo que la necesidad de gestionar con mayor eficiencia las horas – hombre, resulta una oportunidad para ofrecerles mejores servicios en esta área.

#### **4.2.3. Debilidades**

- Falta de Recursos especializados para ciertos proyectos.
- Retrasos en la implementación del proyecto para darles prioridad a otros.
- Renuencia de ciertos líderes de proyectos para adecuarse al cambio.

#### **4.2.4. Amenazas**

- Entes gubernamentales que puedan influir en la ejecución del proyecto.
- El tiempo de implementación de la nueva planificación, gestión y control de horas – hombre, puede no ser suficiente o estar a la medida del tiempo requerido para la implementación de los proyectos, puede entonces, que retrase el avance de los proyectos.
- Se debe lidiar con la rotación de personal y cambio en las políticas internas de la empresa, pues esto significaría un retraso en la implementación de la nueva gestión de horas hombre.

### **4.3. Caracterización del mercado**

#### **4.3.1. Estado actual**

El mercado que maneja Controlca es aquel conformado por empresas de diversos rubros como la banca, el sector empresarial, de consultoría, el sector industrias básicas, el sector de comunicaciones, el sector petrolero, sector educativo y sector de clínicas.

En la actualidad el Mercado de Controlca, se ve inmerso en una diversidad de actores que a su vez se ven afectados por los cambios en material política, que cada



día más merman en las operaciones de todas las empresas de todos los sectores a nivel nacional. Estas volátiles condiciones geopolíticas que presenta el país complica la evolución de este mercado.

A pesar de las dificultades, hasta el momento, Controlca ha apoyado a Inelectra en la construcción de más de 20 plantas de generación de electricidad para el sector eléctrico y la industria petrolera. Estos proyectos incluyen las dos mayores plantas termoeléctricas en operación en Venezuela con una capacidad instalada de 3.800 MW.

#### **4.3.2. Tamaño**

Específicamente, en el caso de Controlca, el tamaño del mercado que maneja actualmente es de unas 20 empresas a las cuales les presta servicios.

#### **4.3.3. Segmentación**

Los segmentos de mercado que atiende Inelectra están dirigidos hacia las siguientes áreas:

- Facilidades de producción de crudos y gas
- Producción y procesamiento de gas
- Refinación y mejoramiento de crudos
- Petroquímica
- Transporte, distribución y almacenamiento de hidrocarburos
- Instalación de plataformas costafuera
- Generación y transmisión de electricidad

#### **4.3.4. Localización**

La localización del mercado que atiende Controlca es en todo el territorio venezolano.

## **5. EL PROYECTO**

El requerimiento que presenta Controlca y que por ende, es lo que se busca en este proyecto, consiste en conseguir un manejo eficiente de los proyectos de la empresa y de los recursos asignados a los mismos, así como la optimización de la asignación de tareas al recurso humano capacitado en base al perfil de conocimiento de los mismos.

Así mismo el proyecto busca, entonces, fidelizar a su cliente directo Inelectra, entregándole productos de alta calidad, a menores costos y en el menor tiempo posible.

Para ello, se debe optimizar los procesos de planificación y gestión de proyectos de ingeniería en telecomunicaciones, haciendo énfasis en la planificación más que en la gestión.

Actualmente, el proceso de planificación de proyectos de consultoría en el área de las telecomunicaciones, se realiza de forma manual y en base a criterio experto. En particular, en Controlca, los responsables del control y gestión de proyectos, llevan a cabo esta tarea de forma “intuitiva”, basados principalmente en su experiencia y formación (criterio experto), lo cual conlleva a un desaprovechamiento en el uso de los recursos disponibles para el desarrollo de actividades, lo que, a su vez, genera gastos innecesarios dada la necesidad de contratar y entrenar personal para tareas específicas durante periodos de tiempo determinados, aun cuando existe un personal capacitado con disponibilidad horaria para el desarrollo de dichas actividades.

### **5.1. Planteamiento Estratégico**

La competitividad y el dinamismo de los mercados actuales obligan a las empresas a estar en una permanente búsqueda de ventajas competitivas sustentables en el tiempo. Destacados autores han llegado a la conclusión de que dichas ventajas pueden obtenerse desde dos perspectivas: la efectividad operacional y la creación de valor único para el cliente. Sin duda, cualquiera sea la perspectiva elegida (o una mezcla de ambas), se requerirá diseñar estructuralmente el negocio de una forma

eficiente y eficaz, que permita obtener las ventajas sostenibles necesarias en forma oportuna.

La visión estratégica del negocio es un pilar fundamental al momento de considerar las motivaciones de un proyecto de desarrollo de ingeniería, particularmente en el área de las telecomunicaciones. Ésta se define como el estado futuro que se espera que la empresa alcance al presentar un desempeño del negocio sustentable, que signifique un incremento en el valor a los clientes y accionistas, por sobre los de la competencia.

Considerando, entonces, la importancia de la visión estratégica del negocio, cabe destacar que la principal motivación para el presente desarrollo, consiste en la búsqueda de una planificación eficiente de los proyectos de ingeniería en el área de las Telecomunicaciones. Dicha planificación es un factor clave a considerar por las empresas, para llevar a adelante y con éxito la gestión y ejecución de cada proyecto.

Para lograr una planificación eficiente, toma fuerza la necesidad de buscar herramientas y metodologías que apoyen el desarrollo de proyectos desde su inicio hasta su fin, tomando en cuenta la información existente en desarrollos de proyectos anteriores, como lo es la estimación de horas para el desarrollo de cada proyecto.

Pero no sólo tomar en cuenta la información de proyectos precedentes basta. Es necesario tener un control adecuado de cada proyecto, que entregue un acceso expedito tanto a los jefes de proyecto, como al responsable general de proyectos, y a los recursos asignados a dichos proyectos. Para, de esta manera, lograr interactuar entre ellos, comunicando los grados de avance de cada tarea, o siendo avisados oportunamente, en caso de un atraso en alguna tarea, o de nuevos requerimientos del proyecto, entre otros.

Esto permite la gestión integral de un proyecto, con lo cual disminuye considerablemente la improvisación y el desconocimiento del estado de las tareas, debido a demoras, problemas con los recursos, aumento excesivo de los costos, etc.

En concreto, la motivación del presente trabajo consiste en el rediseño de los procesos de planificación y gestión de proyectos de Telecomunicaciones, dada una estimación eficiente de horas-hombre y una planificación óptima que tome en cuenta el nivel de experticia del recurso humano así como su disponibilidad horaria.

Todo esto, a través de la optimización del manejo de recursos, por medio de un sistema de planificación y asignación, que simplifique el inicio, progreso y culminación de proyectos de telecomunicaciones.

La importancia de este trabajo, radica en el desarrollo de un sistema de planificación y gestión eficiente capaz de apoyar tanto al responsable de proyectos, como a los jefes de los proyectos respectivos. Lo cual conlleva a mejorar la coordinación con los recursos encargados de llevar a cabo cada proyecto, logrando una disminución en los tiempos de planificación, diseño y evaluación de proyectos, disminución de costos, entre otros.

#### **5.1.1. Planteamiento estratégico, tipo de diferenciación y ventajas competitivas**

La metodología que se utilizará en este proyecto, es la Ingeniería de Negocios dado que “provee los fundamentos y la metodología que permiten diseñar una empresa, incluyendo su arquitectura empresarial, de la cual es parte la arquitectura de procesos, y el detalle de todos los procesos necesarios para que la empresa sea competitiva”<sup>2</sup>.

Es evidente que, en un ambiente globalizado y altamente competitivo, como el que enfrentan las empresas hoy en día, éstas requieren ventajas sustentables en el tiempo, las que pueden lograrse a través de efectividad operacional o posicionamiento estratégico. Ahora, dado que la efectividad operacional sólo genera ventajas competitivas en el corto plazo, y el objetivo del proyecto es generar mejoras

---

<sup>2</sup> Barros, O. Ingeniería e-Business: Ingeniería de Negocios para la Economía Digital. J.C. Sáez Editor, 2004.

sustentables en el tiempo y con un enfoque sistémico, se utilizará el posicionamiento competitivo como tipo de ventaja competitiva.

Sabiendo que el posicionamiento debe involucrar actividades que sólo son propias de la empresa de forma exclusiva, se define el posicionamiento de este proyecto como posicionamiento por necesidad, ya que se busca satisfacer todas las necesidades del cliente.

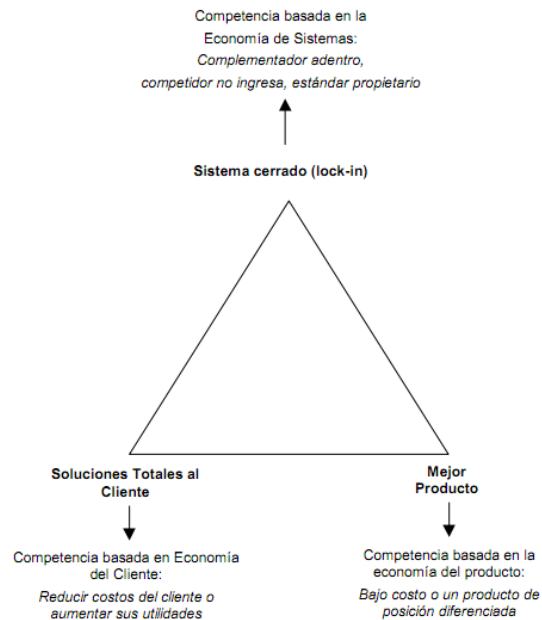
De esta forma, una vez definido cuál es el posicionamiento, podemos entrar a responder cuál es la estrategia, la cual está enmarcada en la comprensión y relación íntima con el cliente, para así entregar productos de alta calidad, coherentes al cumplimiento de sus necesidades. Estos productos apuntarán a mejorar la capacidad del cliente para que éste pueda crear su propio valor económico. Dicha estrategia está definida como Solución Integral para el Cliente.

### **5.1.2. Posicionamiento Estratégico según Hax**

Según las ideas de Hax<sup>3</sup> una empresa se debe a sus clientes. El centro de la estrategia debe ser el cliente, intentando atraerlo, satisfacerlo y retenerlo. Dentro de ese espíritu, se establecen 3 visiones las cuales no son 100% excluyentes pero se estudian de esta manera para su mejor comprensión. Estas visiones se ven representadas en la siguiente figura.

---

<sup>3</sup> Arnoldo C. Hax: Profesor de Technological Innovation, Entrepreneurship, y Strategic Management en el Instituto Tecnológico de Massachusetts (MIT)



La estrategia Solución Integral para el Cliente, definida para este proyecto en particular, se ha escogido ya que en el Modelo de Negocios actual se ha logrado que Controlca esté inserto en la cadena de valor del cliente, al tomar las actividades de planificación, programación y control de proyectos de telecomunicaciones.

### 5.1.3. Solución integral al cliente

La idea aquí es cambiar el paradigma: en vez de enfocarnos en superar a nuestros competidores, nos enfocamos en profundizar la relación con nuestros clientes ofreciéndoles un conjunto más completo de productos y servicios que incrementa el valor que perciben. La transformación hacia la solución integral al cliente requiere de una manera diferente de capturarlo. Así es como se trabaja al menos en 2 conceptos relacionados como lo son:

1. **Integración con el cliente:** esto corre por cuenta de la facilitación de actividades, las que son tomadas por Controlca y ejecutadas tal como si fuera el cliente quien las realizara, para quien es transparente este proceso. Lo importante aquí es obtener una buena definición del proceso que se realizará, para ejecutarlo correctamente y lograr que el cliente aprecie el valor que esto significa.

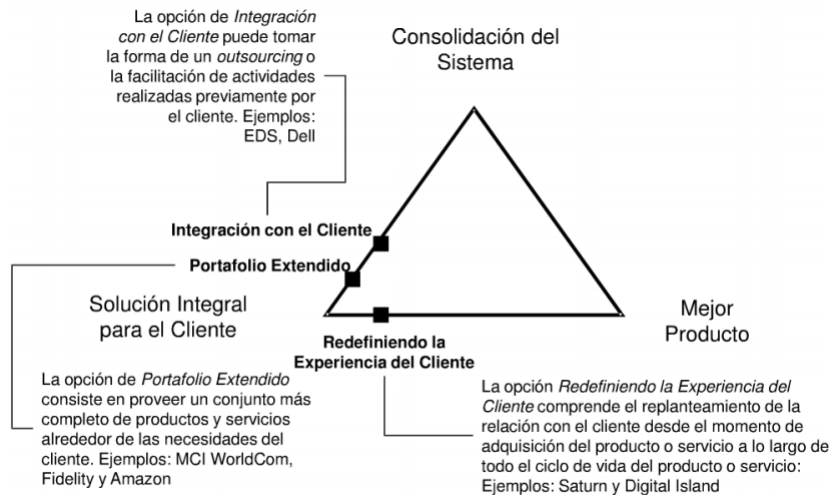
2. **Portafolio Extendido:** esta visión consiste en extender el conjunto de servicios alrededor de las necesidades del cliente. En este caso, Controlca comienza con el control y gestión de proyectos de telecomunicaciones para Inelectra, pero implementa en sus procesos tareas de asesoramiento a las decisiones gerenciales, desarrollando un rol de asesor tecnológico, basado en su experiencia.

Hax postula también que para aplicarse esta visión desde el inicio las empresas deben saber cómo centrarse en el cliente y ofrecerle una propuesta de valor que tenga cuatro atributos, con los cuales cumple este proyecto:

1. **Que sea única:** la implantación de proyectos de telecomunicaciones, desarrollados en su totalidad por Controlca hacen que esta solución sea particularmente específica para las necesidades únicas de este cliente en particular.
2. **Que no sea fácil de imitar:** dadas las razones del punto anterior, los más de 25 años de trabajo profesional en el ámbito de las telecomunicaciones y de las ventajas competitivas con que cuenta Controlca para el desarrollo de de forma rápida y eficiente de proyectos en esta área tan específica de la consultoría hacen que esta solución sea muy compleja y difícil de imitar e inclusive de reemplazar por parte de Inelectra.
3. **Que agregue valor:** producto de la integración con el cliente es que éste percibe el valor, no solo en el desarrollo de proyectos de telecomunicaciones, sino también valoran el tiempo que se ahorran al desarrollar proyectos de este tipo, así como la alta calidad de los productos entregados, resultando en los tiempos planificados.
4. **Que produzca un lazo inquebrantable con el cliente:** justificado por la alta calidad y la respuesta inmediata en el desarrollo de soluciones de telecomunicaciones ideales para el cliente final son las que producen un

lazo inquebrantable, pues se ha trabajado en una relación basada en la confianza y en los buenos resultados que los desarrollos realizados por Controlca han logrado.

Una representación gráfica de esta visión, se puede apreciar en la siguiente figura.



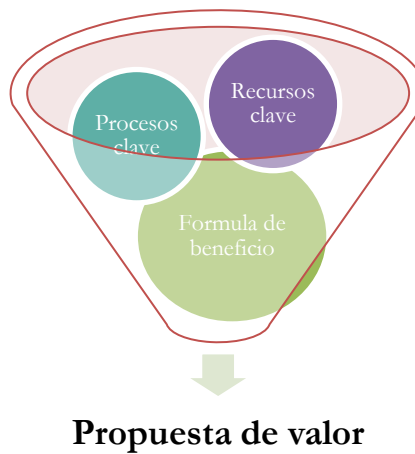
## 5.2. Modelo de Negocios

En la siguiente sección se analizarán las diferentes variables que componen el modelo de negocios propuesto, tomando en cuenta la intención de reducir los costos totales inherentes al proyecto. Así como mejorar y optimizar la planificación y gestión de proyectos de telecomunicaciones.

Para el desarrollo de este modelo de negocios se tomó como marco de referencia el artículo “Reinventing your business model”, del autor Henning Kagermann, en el cual se presentan 4 elementos que en forma conjunta crean y entregan valor.

Estos elementos son: Propuesta de valor para el cliente, Fórmula de beneficios, Recursos Claves y Proceso claves.





**Propuesta de Valor:** las propuestas de valor a los clientes representan los atributos que las empresas suministran a través de sus productos y servicios para crear fidelidad y satisfacción en los segmentos de clientes seleccionados. La propuesta de valor es un concepto clave para poder identificar los procesos internos, del negocio, establecer los inductores e indicadores y la infraestructura necesaria que le dará vida a la estrategia.

**Recursos Claves:** los recursos claves, son activos tales como las personas, tecnología, productos, instalaciones, equipos necesarios para entregar la propuesta de valor al cliente.

**Procesos Claves:** son aquellos procesos que permiten generar valor de forma satisfactoria.

**Fórmula de beneficios:** es el modelo que define cómo la empresa crea valor para sí misma y beneficia al cliente.

A continuación se define el producto y el cliente al que va dirigido.

### **Producto**

- Los documentos y planos de los Sistemas de Telecomunicaciones y Vigilancia Electrónica
- Libros de ingeniería

- Requisiciones de materiales
- Análisis técnicos

### **Cliente**

- Para el producto anteriormente descrito se tienen el siguiente cliente:  
Inelectra

### **5.3. Propuesta de valor**

La Propuesta de valor consiste en entregar una solución a un cliente con un problema dado en una situación dada o satisfacer una necesidad en específica. De esta forma, y con el objeto de presentar una solución completa en integral al cliente, se define la propuesta de valor como: entregar Proyectos de Telecomunicaciones de alta calidad a bajos costos.

Esta propuesta de valor se logrará a través de la determinación eficiente de horas-hombre, planificación de proyectos, asignación optimizada y gestión de recursos, en forma rápida, eficiente, y cada vez con un mayor grado de consistencia y solidez.

#### **5.3.1. Variables a considerar**

Para poder cumplir con lo estipulado en la estrategia, es de suma importancia determinar el valor de horas-hombre adecuado por producto de telecomunicaciones a ser desarrollado en el proyecto. Aunado a esto es necesario el desarrollo de un sistema integrador que planifique, asigne y gestione, en paralelo con un rediseño a los procesos de gestión y desarrollo de la organización, en los que implica la creación de nuevas metodologías a seguir por parte de los ingenieros y jefes de proyecto encargados de llevar a cabo los mismos.

Se deberá tener una especial preocupación por los tiempos definidos para cada tarea, para lo que será necesario establecer una medición de los tiempos de implementación de cada una de ellas y cómo éstos varían de una implementación a

otra, tratando de seccionar cada tarea lo más que sea posible, a modo de poder establecer tiempos de desarrollo más fidedignos, ya que si una tarea es muy específica y puntal, es más fácil de determinar su duración que si fuese más general y poco acotada.

Como consecuencia de la determinación adecuada de las horas-hombre y la asignación optimizada de tareas a recursos, se generará una disminución proporcional de los costos totales asociados al proyecto, lo cual implica una medida del grado de satisfacción del cliente.

#### **5.4. Recursos Claves**

Para el desenvolvimiento efectivo del proyecto, se deberá tomar en cuenta los siguientes recursos claves que contribuyen a generar la propuesta de valor al cliente:

- **Líderes de proyectos**

Es el recurso humano indispensable y adecuado para poder impulsar de forma exitosa la solución en la organización. Son las personas adecuadas para promover el cambio en el modelo paradigmático que implica el desenvolvimiento laboral de los ingenieros participantes de cada uno de los proyectos.

- **Tecnología**

La tecnología habilitante que asistirá la solución será Microsoft.Net Framework, que hace énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

- **Información**

Se necesita comunicar de manera clara y transparente, así como de forma asertiva y oportuna todos los cambios que la solución implica, a todo el recurso humano que participará de la misma, ésto con el objetivo de manejar una misma información sin ambigüedades, lo que permite una mejor cohesión en los equipos que desempeñan las actividades en cada proyecto.

## **5.5. Procesos Claves**

El proceso primordial que se verá afectado es el de preparación y ejecución de proyectos de telecomunicaciones, específicamente lo relacionado con la creación, planificación y control de los mismos.

Este punto será tratado en la sección 5 del presente informe con mayor detalle.

## **5.6. Beneficios esperados**

Los beneficios esperados una vez que el sistema este implementado son:

### **a) Aprovechar la información de proyectos anteriores para el desarrollo de futuros proyectos**

Ésto como apoyo a la planificación, debido a que tanto el responsable de proyectos como los jefes de proyectos, encargados del ingreso de nuevos requerimientos, definen las actividades, plazos, y recursos a utilizar, entre otros, basándose en su criterio y experiencia, pero desaprovechando en muchos casos la información de una gran cantidad de proyectos previamente realizados.

### **b) Utilizar un conocimiento más acabado de las capacidades de cada recurso**

Uno de los problemas más recurrentes en Controlca, actualmente, es el poco conocimiento de las habilidades de su recurso humano. Asignando al personal capacitado a tareas específicas, se podrán ejecutar tareas en menor tiempo y de mejor calidad.

### **c) Disminución de tiempos de diseño, evitando el “recurso ocioso”:**

Con la implementación de la solución, se pretende poder estimar de mejor forma los tiempos de desarrollo, lo que quiere decir que cada ingeniero podrá desarrollar las tareas más eficientemente, aprovechando mejor los tiempos.

#### **5.6.1. Cuantificación de los Beneficios**

Se estima que, con la realización de este proyecto, las variables para cuantificar sus beneficios se verán reflejadas en los tiempos de desarrollo de productos y costos asociados a los mismos, así como la precisión en la planificación y asignación de recursos.

##### **Directos**

- Disminución de tiempos de planificación y mayor precisión en la definición de plazos y costos de las actividades a realizar
- Disminución de los costos totales inherentes al proyecto (impacto directo en el cliente)
- Mayor precisión en la asignación de recursos (menos contrataciones)

##### **Indirectos**

- Aumento de los niveles de satisfacción de los clientes.
- Disminución de los tiempos de desarrollo de los proyectos, lo que se traduce en una disminución en los costos.

#### **5.7. Definición del Proyecto**

La Empresa desea conseguir un manejo eficiente de sus proyectos y de los recursos asignados a los mismos, así como la optimización de la asignación de tareas al recurso humano capacitado en base al perfil de conocimiento de los mismos.

Así mismo el proyecto busca fidelizar a su cliente directo, Inelectra, entregándole productos de alta calidad, a menores costos y en el menor tiempo posible.

Para ello debe optimizar los procesos de planificación y gestión de proyectos de ingeniería en telecomunicaciones, haciendo énfasis en la planificación más que en la gestión.

### **5.7.1. El problema**

Actualmente, el proceso de planificación de proyectos de consultoría en el área de las telecomunicaciones, se realiza de forma manual y en base a criterio experto. En particular, en Controlca, los responsables del control y gestión de proyectos, llevan a cabo esta tarea de forma “intuitiva”, basados principalmente en su experiencia y formación (criterio experto), lo cual conlleva a un desaprovechamiento en el uso de los recursos disponibles para el desarrollo de actividades, lo cual a su vez genera gastos innecesarios dada la necesidad de contratar y entrenar personal para tareas específicas durante periodos de tiempo determinados, aun cuando existe un personal capacitado con disponibilidad horaria para el desarrollo de dichas actividades.

### **5.7.2. Objetivos estratégicos**

En concordancia con lo anterior, es posible describir un primer objetivo estratégico del proyecto: proveer al cliente de productos de Diseño y Desarrollo de Sistemas de Telecomunicaciones de alta calidad, a costos totales más bajos a través de la diferenciación que representa la gestión eficiente del tiempo utilizado para desarrollar dichos productos.

Este primer objetivo implica la determinación acertada del cálculo de las horas-hombre necesarias de cada producto para completar con éxito cada una de las tareas específicas, a través de ciertos criterios de estimación de diseño que miden la complejidad del proyecto.

El segundo objetivo estratégico del proyecto consiste en: generar una óptima planificación de horas-hombre a lo largo del desarrollo de cada proyecto en base a la

experiencia, destreza y habilidades de los recursos disponibles, asegurando su disponibilidad horaria para el exitoso desarrollo de las tareas.

Finalmente, el último objetivo estratégico pertinente a este proyecto es: gestionar la planificación del proyecto, específicamente la administración de horas-hombre, estado de avance de productos, atrasos, entre otros indicadores, de forma eficiente que entregue un mayor grado consistencia de la información para proyectos futuros.

Estos objetivos entregan las bases para definir las métricas que permitirán cuantificar el éxito del proyecto. Éstas permitirán desarrollar los conceptos de un cuadro integral de mando, el cual considerará las siguientes estrategias e iniciativas:

**Perspectiva financiera:** crecer con rentabilidad sustentable.

- Incrementar y consolidar la participación de Controlca en proyectos de ingeniería en el área de telecomunicaciones de Inelectra, tanto a nivel nacional como internacional.
- Mejorar el costo total inherente al proyecto, a través de la estimación adecuada de las horas hombre y asignación eficiente de tareas a recursos.

**Perspectiva clientes:** ser reconocido como el mejor prestador de servicios profesionales de ingeniería en el área de telecomunicaciones.

- Incrementar el grado de satisfacción del cliente
- Incrementar el número de licitaciones adjudicadas

**Perspectiva procesos internos:** mejorar la excelencia operacional de la empresa con altos estándares de calidad

- Mejorar la calidad y el ciclo de entrega de los diseños de ingeniería
- Optimizar la planificación de desarrollo de productos
- Perfeccionar la asignación de horas-hombre
- Gestionar de forma eficiente los recursos de los proyectos

### **5.7.3. Evaluación Financiera**

En esta sección se mostrará la evaluación económica que conlleva incorporar la solución propuesta, dados los costos operativos de desarrollar un proyecto cualquiera, los costos de inversión necesarios para la implementación de esta solución y el potencial ahorro que esta solución representaría.

**Tiempos de desarrollo e implementación de la solución:** equipo requerido para el desarrollo del Proyecto.

- Se considerarán dos Jefes de Proyectos para servir de apoyo al momento de implementar la solución. Se destinarán un total de 10 horas semanales para la revisión de los avances del proyecto por 5 meses.
- Además del tiempo necesario para llevar a cabo lo anterior, se destinarán dos Ingenieros de Proyectos para apoyo en el diseño, programación y puesta en marcha de la solución. Se destinarán 30 horas a la semana por 5 meses.

#### **Costos de desarrollo e implementación de la solución**

- Inversión Inicial: la compra de un servidor de aplicaciones y de bases de datos, representa una inversión inicial de aproximadamente 5.000 Bs.
- Costo de hora hombre del Jefe de Proyectos: 200 Bs.
- Costo de hora hombre del Ingeniero de Proyectos: 100 Bs.



A continuación se muestran los costos por llevar a cabo un proyecto expresados en moneda local, el Bolívar.

	N° Horas x Día (promedio)	Total de Horas Semanales	Total de Semanas	Costo HH	Costo Total
Jefe de Proyecto 1	2	10	20	Bs. 45,00	Bs. 9.000,00
Jefe de Proyecto 1	2	10	20	Bs. 45,00	Bs. 9.000,00
Ingeniero de Proyecto 1	6	30	20	Bs. 25,00	Bs. 15.000,00
Ingeniero de Proyecto 2	6	30	20	Bs. 25,00	Bs. 15.000,00
Servidor					Bs. 5.000,00
					<b>Bs. 53.000,00</b>

El Bolívar, tiene una equivalencia aproximada con el Peso Chileno de 50 CLP/Bs., por lo que la inversión en Pesos Chilenos equivale aproximadamente a CLP 2.650.000,00

#### 5.7.4. Plan de Marketing

Para dar respuesta a las necesidades actuales que posee Controlca en el área de su gestión de horas – hombre, se define el siguiente plan de marketing que pretende vender el presente proyecto a la mencionada empresa.

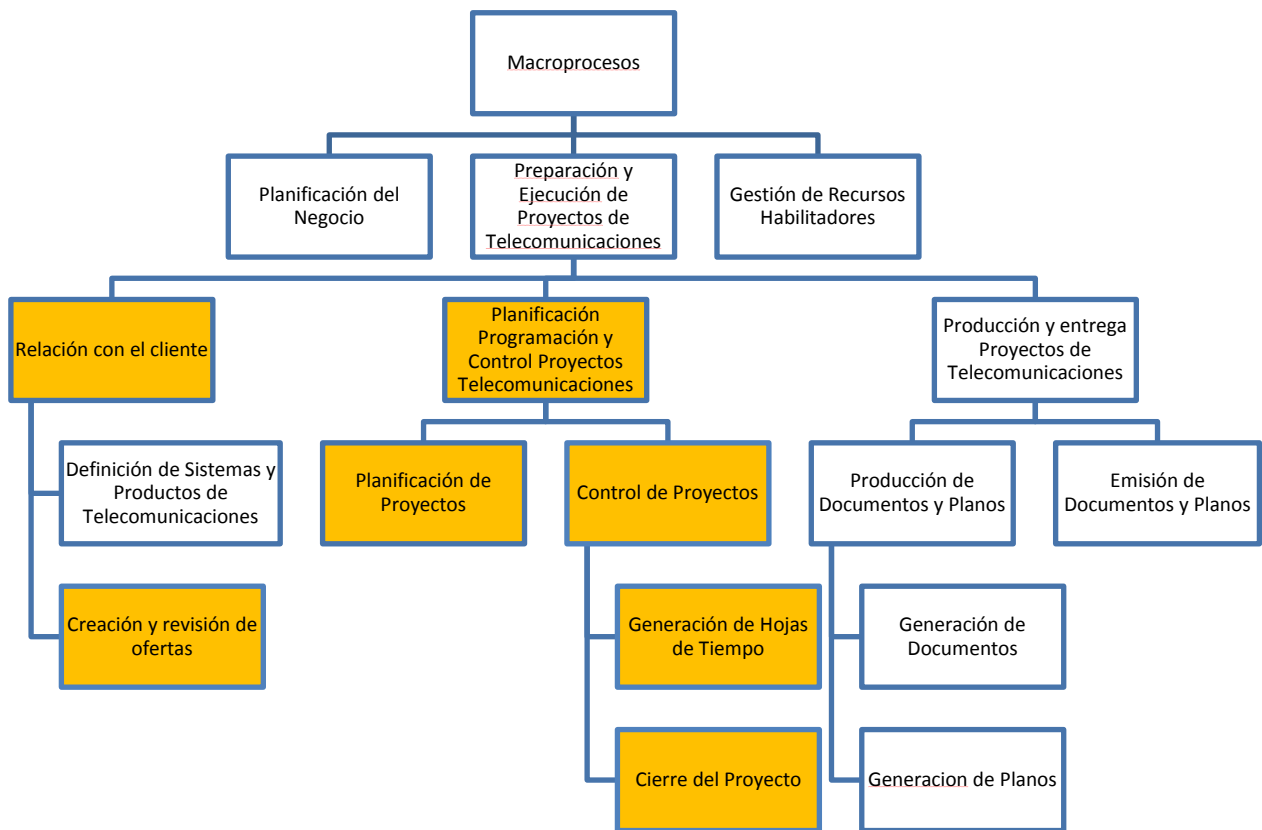
- **Objetivo:** Concretar la implementación de la Optimización en el Proceso de Gestión de Proyectos de Consultoría de Ingeniería
- **Estrategia:** Abordar a los diversos stakeholders decisivos en la implementación de soluciones de gestión de horas – hombre, a fin de lograr la aplicación del presente proyecto.
- **Tácticas:**
  1. Coordinar reuniones con los directivos de Controlca a fin de darles a conocer el proyecto

2. Dirigir workshops dedicados a los líderes de proyectos a fin de que conozcan la implementación del proyecto
  3. Establecer una pauta de preguntas y respuestas frecuentes que puedan ser abordadas en las reuniones y workshops, para esclarecer todo tipo de dudas sobre la implementación del proyecto.
  4. Coordinar una reunión para discutir el presupuesto con el cual se manejará la venta del proyecto.
- **Evaluación y Control:** En esta etapa se definirá un modelo de evaluación a la implementación del proyecto a fin de conocer y manejar las fallas que este pueda tener al momento de la ejecución. Así mismo, se evaluará el nivel de satisfacción que presenta el proyecto una vez implantado.

## 6. REDISEÑO DEL PROCESO DE NEGOCIOS Y APOYO COMPUTACIONAL

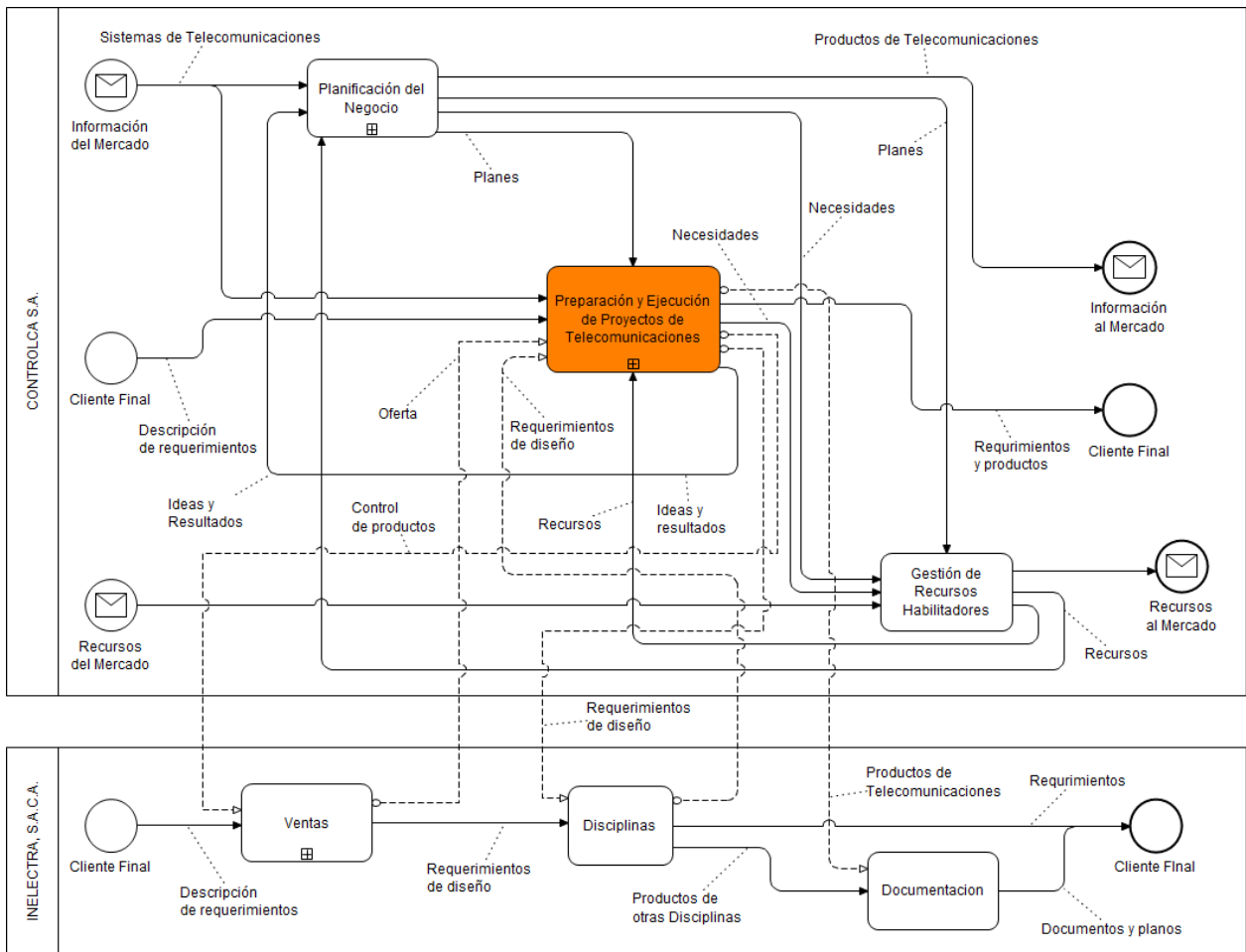
En el presente capítulo se detalla el trabajo de rediseño realizado y se definen las actividades que requieren apoyo computacional. En particular, el proyecto se enfoca en el rediseño de los procesos relacionados con creación de oferta, planificación, asignación de recursos y control de proyectos, detallando cuál es la lógica diseñada para estos efectos.

A continuación se presenta la el rediseño de Controlca representada mediante un árbol de procesos, indicando el color aquellos sobre los cuales se hará un rediseño de los procesos.



A continuación se presenta la estructuración de la arquitectura de macroprocesos para el rediseño de la solución planteada en el marco de los proyectos desarrollados al interior de Controlca.

Cabe resaltar que el código de colores usado en los siguientes diagramas de procesos, es sólo referencial y sirve para resaltar los puntos más relevantes de los mismos.



En esta arquitectura de procesos, el macroproceso más relevante para el modelo de negocios propuesto es **“Preparación y Ejecución de Proyectos de Telecomunicaciones”** (Macro 1) , de la cual se describirán cada uno de los subprocesos relevantes en el rediseño. Este macroproceso representa la cadena de valor de Controlca, ya que en éste, es donde se generan los diferentes productos que ofrece la empresa a su cliente, Inelectra.

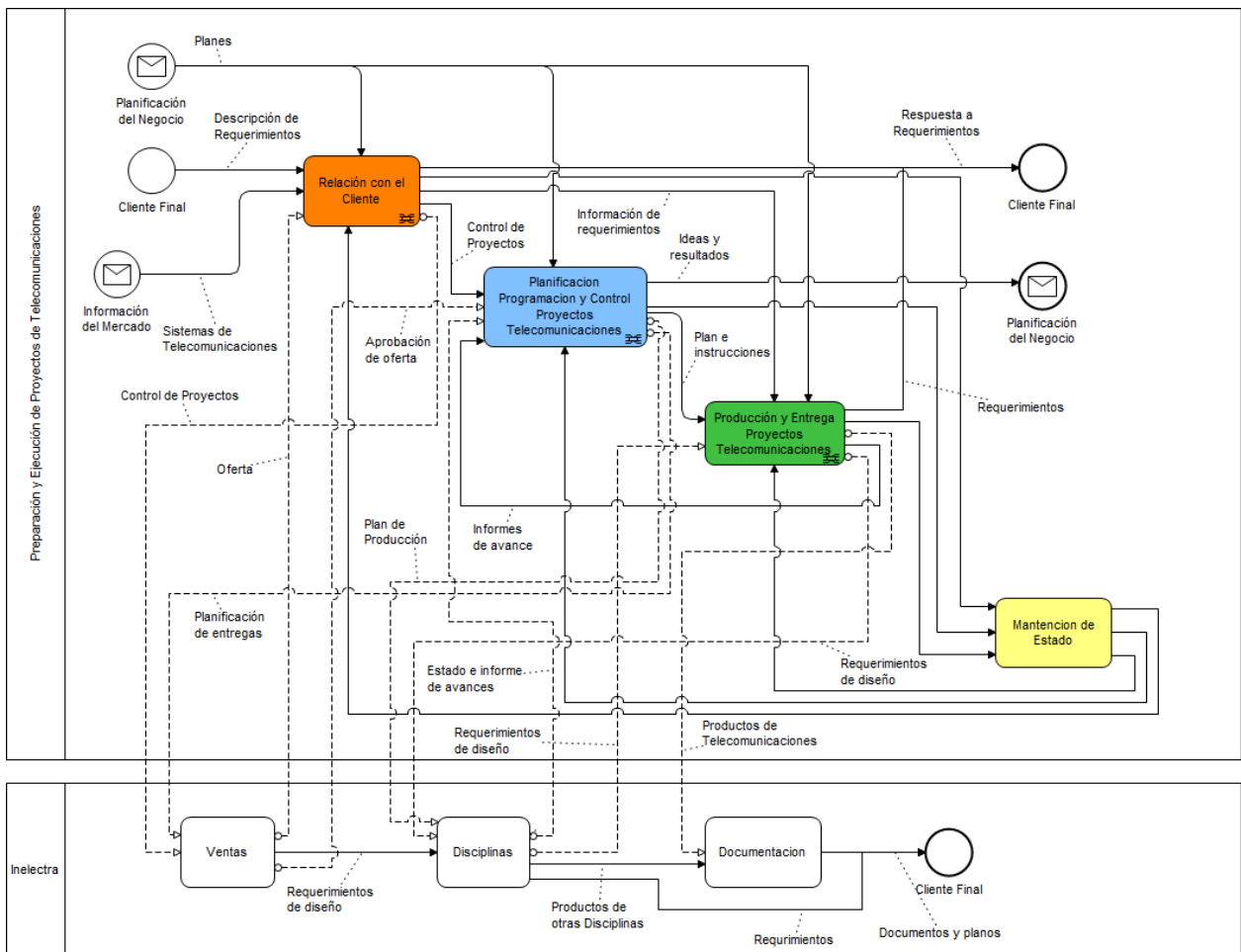
El macroproceso **“Planificación del Negocio”** (Macro 3) comprende el diseño y la elaboración del Plan de Desarrollo Estratégico de Controlca, para organizar internamente el establecimiento y asignar las tareas correspondientes.

El macroproceso de **“Gestión de Recursos Habilitadores”** (Macro 4) tiene como propósito estimar los recursos que necesita Controlca y asegurar que éstos sean provistos, decidiendo, por ejemplo, contratar más personal, ingenieros o técnicos,

especialistas en el área de las telecomunicaciones y afines, adquirir nuevos software de desarrollo, nuevos equipos, entre otros, en fin, todo lo que sea recurso habilitante para desarrollar los documentos y planos de forma efectiva, eficiente y de calidad.

Se puede observar en la Macro 1 la interacción entre Controlca y su cliente Inelectra durante la etapa de preparación y ejecución, sobre la cual se trabajara en el presente trabajo de grado.

A continuación se presenta el macroproceso “Preparación y Ejecución de Proyectos de Telecomunicaciones”.



Dentro de la Macro 1 se tienen los siguientes procesos:

**“Relación con el cliente”**: comprende las actividades de análisis que se requieren para inducir y guiar las ventas, así como las actividades de ventas, contactos por servicios al cliente, procesamiento de órdenes y preparación de ofertas, además incluye la decisión de factibilidad y conveniencia de aceptarlas.

**“Planificación, Programación y Control Proyectos de Telecomunicaciones”**: como su nombre lo indica, aquí se realiza la planificación, programación y control requeridos para la generación y entrega de los productos, incluyendo la planificación de capacidad.

**“Producción y entrega Proyectos de Telecomunicaciones”**: lleva a la práctica los planes y programas del proceso anterior, realizando las actividades físicas de generación y distribución de productos.

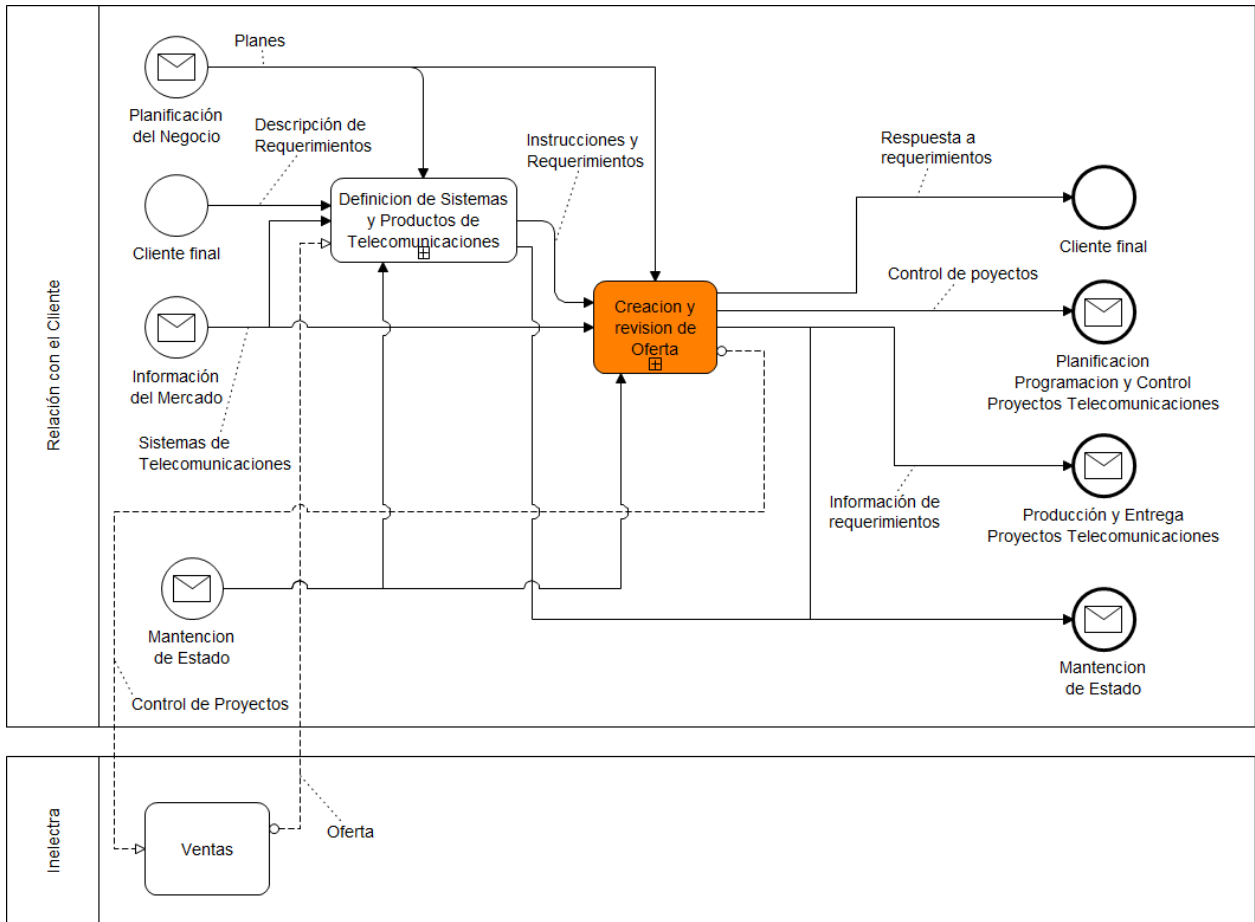
Es importante destacar que se ha incorporado el proceso **“Mantenimiento de Estado”**, el cual no existe formalmente en Controlca y se incluye como parte del rediseño.

Los procesos donde se pretende impactar directamente son los relacionados con la planificación y gestión del proyecto, específicamente en **“Relación con el Cliente”** y **“Planificación, Programación y Control Proyectos de Telecomunicaciones”**.

## **6.1. Relación con el cliente**

Es a partir de este proceso que comienzan a incorporarse modificaciones a la estructura actual o rediseño de procesos.

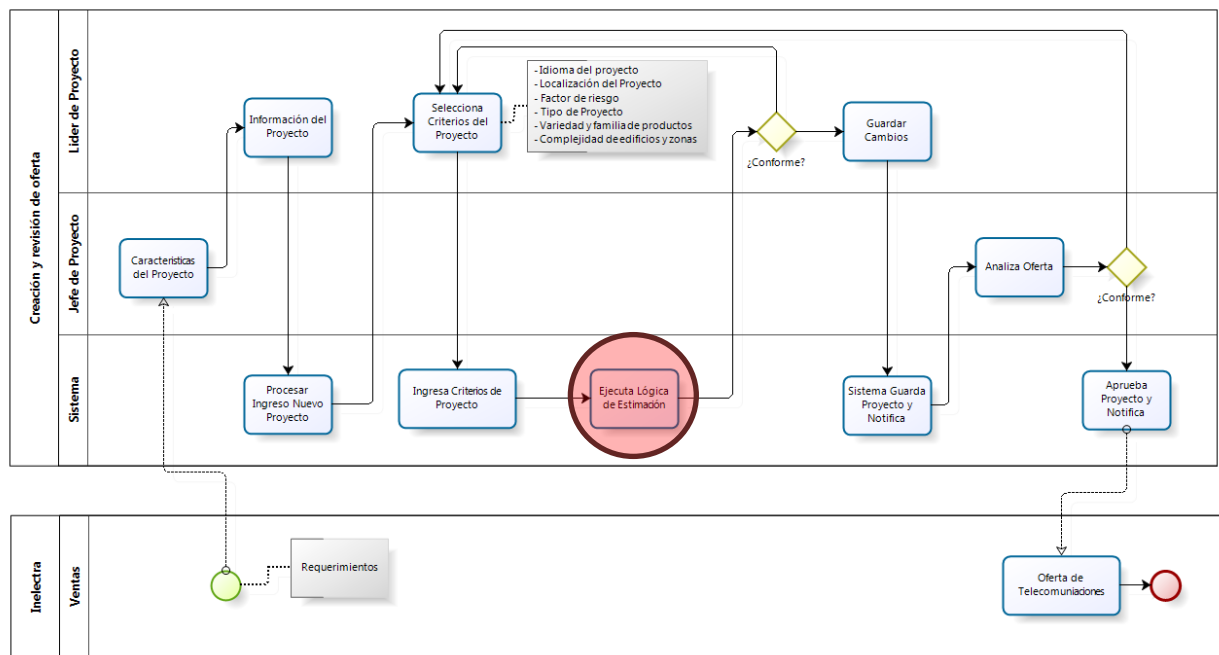
En esta etapa Controlca recibe de Inelectra Ventas, los requerimientos y especificaciones sobre sistemas de telecomunicaciones y vigilancia electrónica que el cliente final solicita. A partir de ese momento, la gerencia general de Controlca, junto a los especialistas de los sistemas de telecomunicaciones y vigilancia electrónica, definen los sistemas y productos que deberán ser desarrollados para cumplir con los requisitos planteados por el cliente final.



En particular, se incorpora la **automatización en la generación de las ofertas**, basada en proyectos antiguos, y la tipificación de los productos, dados ciertos criterios de evaluación y reglas de negocio que afectan el cálculo de horas-hombre requeridos para su desarrollo.

La tipificación de productos será el modelo utilizado para la estimación del tiempo necesario de desarrollo por cada producto. Para cada tipo de producto, dentro de un Sistema de Telecomunicaciones, se definirán variables de cambio dada la sensibilidad de las mismas a las magnitudes de cambio entre un proyecto y otro. Adicionalmente al momento de cierre del proyecto se calculará el error de estimación por cada producto estimado con el fin de utilizarlo para minimizar el error del cálculo en los siguientes proyectos.

A continuación se presenta en forma detallada el proceso de creación de una oferta.



Partiendo de la solicitud por parte de Inielectra se definen las siguientes variables para la estimación del esfuerzo de diseño, las cuales conforman la primera lógica de negocios dentro del sistema de apoyo computacional.

Cabe destacar que los resultados de dicha tipificación serán contrastados con proyectos reales pasados con el fin de determinar el nivel de error de la estimación. Esta etapa será presentada en la siguiente presentación del curso.

Variables para la estimación del esfuerzo de diseño:

- a) **Idioma del proyecto:** el idioma base del proyecto será Español, con la posibilidad de desarrollar proyectos en Inglés, lo cual agrega un cierto grado de complejidad que hay que tomar en cuenta.
- b) **Localización del Proyecto:** se tomará como referencia base, el territorio nacional, dadas las facilidades de traslado. En caso de ser un proyecto con sede internacional, se deberá complejizar el cálculo debido a la necesidad de hacer estimaciones de traslado, inspecciones, informe de visita, entre otros.



- c) **Factor de riesgo:** hace referencia al tipo de cliente y la experiencia basada en el re-trabajado de productos, de una mala coordinación o cambios de alcance. Puede ser definido como: bajo, medio, alto o muy alto.
  
- d) **Tipo de Proyecto:** indica la variedad y familia de productos dado un tipo de ingeniería a desarrollar. Entre los tipos de Proyecto están:
  - a. Ingeniería conceptual
  - b. Ingeniería básica
  - c. Ingeniería de detalle
  - d. FEED
  - e. IPC
  
- e) **Complejidad de edificios y zonas:** para cada edificio o zona será necesario definir un nivel de complejidad del 1 al 5, donde 1 indica el mínimo valor de complejidad y 5 el máximo. Además habrá que indicar para cada uno de éstos, los sistemas de telecomunicaciones asociados, a fin de ajustar las horas-hombre de los productos a desarrollar.

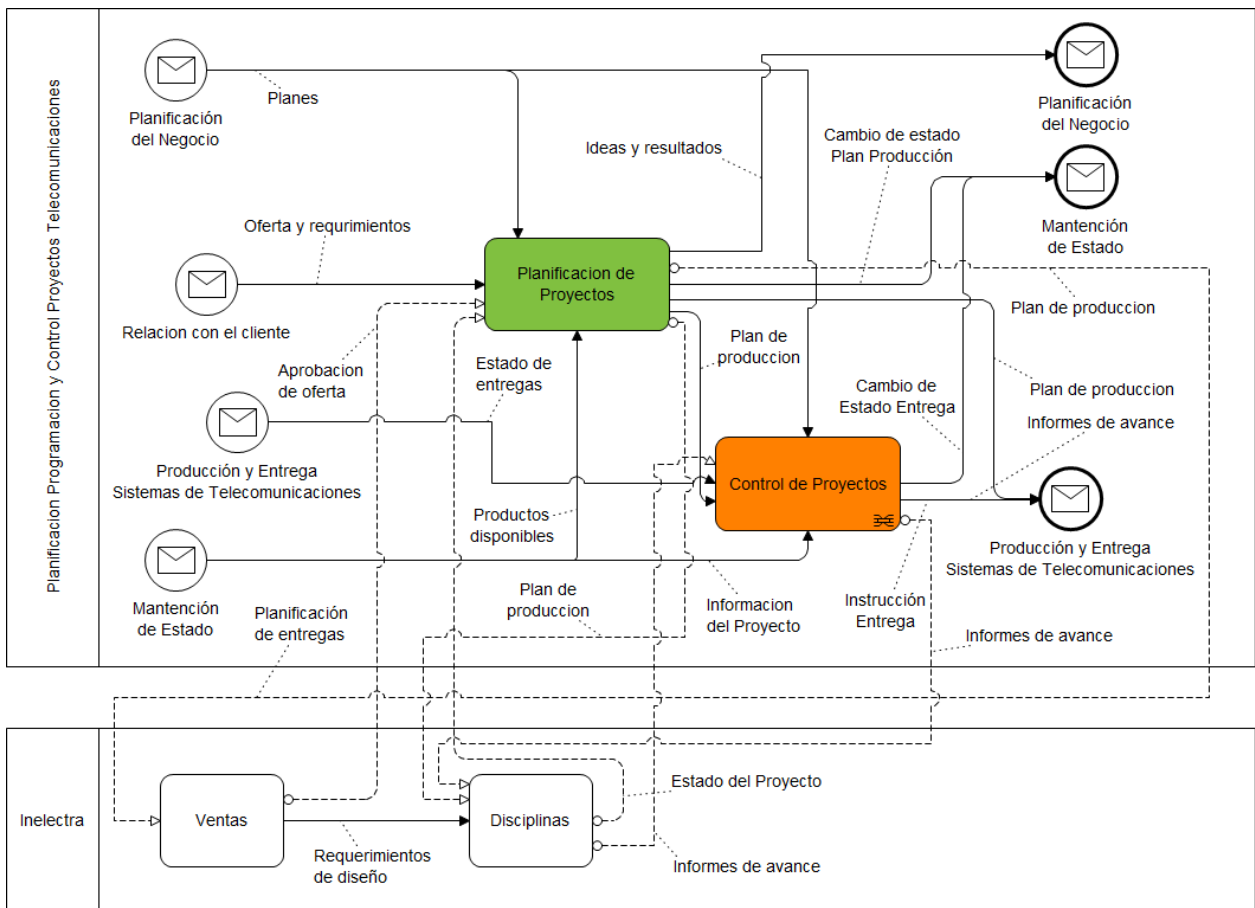
Es importante resaltar que los valores de las variables de estimación variarán significativamente dependiendo del tipo de producto a entregar y el sistema de Telecomunicaciones relacionado.

Es importante destacar que cada producto será definido con un valor base de horas necesarias para su producción. Sobre este valor y definidas las variables de tipificación mencionadas anteriormente se realizará un recargo de horas hombre.

## 6.2. Planificación, Programación y Control Proyectos de Telecomunicaciones

En este proceso se realiza la planificación y el control del proyecto, partiendo con la generación del “Control de Productos”, llamada CDP, el cual refleja el total de horas-hombre por producto a ser asignadas a los ingenieros Lider, Senior y Junior. La asignación de horas-hombre se calcula con base en el ingeniero Junior, quien tiene un valor fijo sobre el total de horas, calculado previamente. El valor de horas otorgado a los ingenieros Senior y Lider, es un valor porcentual del Junior.

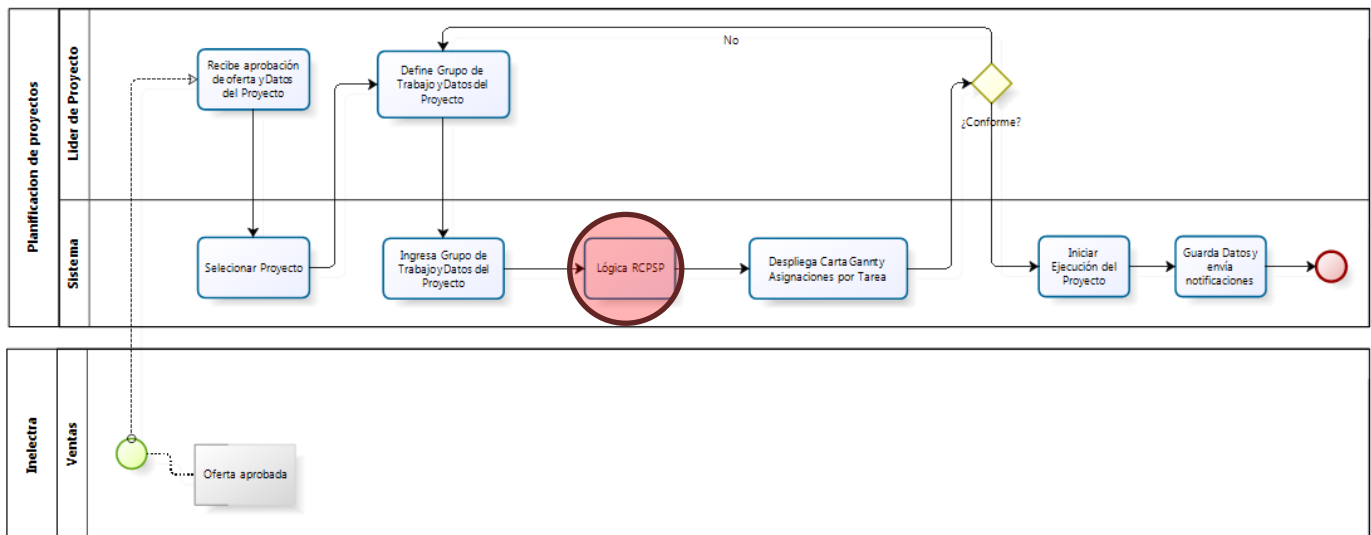
Dependiendo del Proyecto y el producto de Telecomunicaciones a entregar puede que sólo participen el ingeniero Senior y el Lider o únicamente el Lider.



Una vez definida la carga horaria por producto para su diseño y revisión, se definen los equipos de trabajo basado en el perfil del ingeniero, es decir, su experiencia en desarrollo de Sistemas de Telecomunicaciones, específicamente los requeridos por

el proyecto, así como en la disponibilidad de horario que posea, basado en los datos entregados por el sistema, referentes a las asignaciones previamente otorgadas y no finalizadas.

A continuación se presenta en forma detallada el proceso de “**Planificación de Proyectos**” en BPMN, el cual contempla la planificación de entregas y asignación de recursos para el diseño de documentos y planos de Sistemas de Telecomunicaciones.



En este proceso se define la más importante lógica de negocio, la cual requiere de un significativo apoyo computacional.

La lógica RCPSP, es la encargada de realizar la planificación optimizada, así como la asignación y priorización de actividades. Esta lógica hace uso del algoritmo de optimización *Branch and bound*, el cual hace uso de una estructura de árbol construida dinámicamente, que busca representar el espacio de todas las secuencias posibles, que en este caso en particular, deriven en posibles escenarios de planificación. La búsqueda comienza en el nodo raíz y continúa hasta llegar a un nodo hoja.

El algoritmo toma como entradas los siguientes datos:

- a) **Tipo de recursos:** experticia de los ingenieros
- b) **Disponibilidad de cada tipo de recurso:** número de ingenieros disponibles
- c) **Actividades**
  - a. Duración
  - b. Consumo de recurso (cantidad): cuántas horas hombre requiere la actividad
  - c. Actividades predecesoras

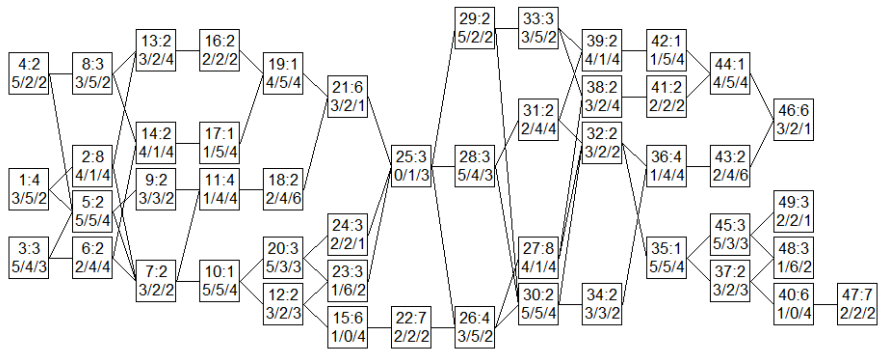
Un detalle indispensable es la definición del “Perfil de ingeniero”, el cual equivale al tipo de recurso según la terminología de algoritmo. El perfil de ingeniero indica las habilidades que posee el ingeniero y refleja su nivel de conocimiento en una escala del 1 al 5. Donde 1 indica el menor nivel de conocimiento. Los niveles de conocimiento agrupan tópicos que el ingeniero domina.

La información sobre el proyecto, actividades, recursos y prioridades, es representada en formato PSPLIB, *Project Scheduling Problem Library*, y es procesada por el algoritmo *Branch and bound* de la herramienta RESCON, desarrollada por Demeulemeester & Herroelen<sup>4</sup>.

Los datos del proyecto son interpretados en forma de grafo, como se presenta a continuación:

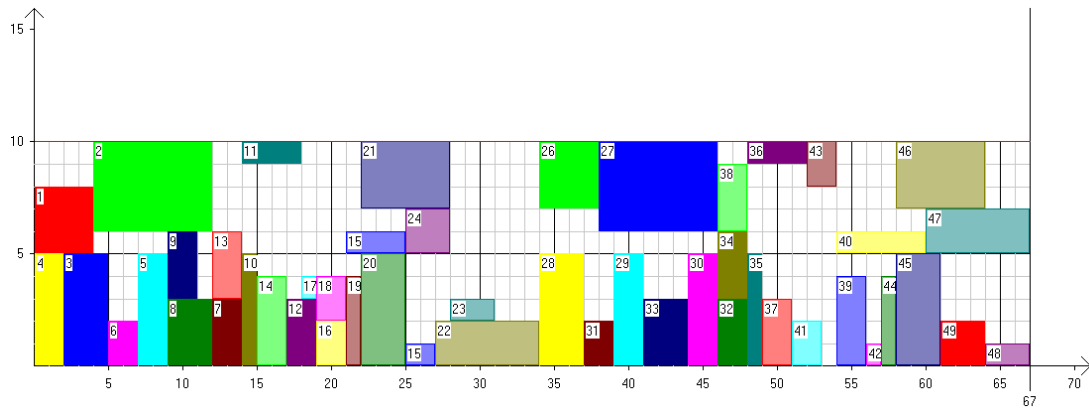
---

<sup>4</sup> Demeulemeester, E., and Herroelen, W., 1992, A branch-and-bound procedure for the multiple resource-constrained project scheduling problem, *Management Science* 38(12), pp 1803-1818.

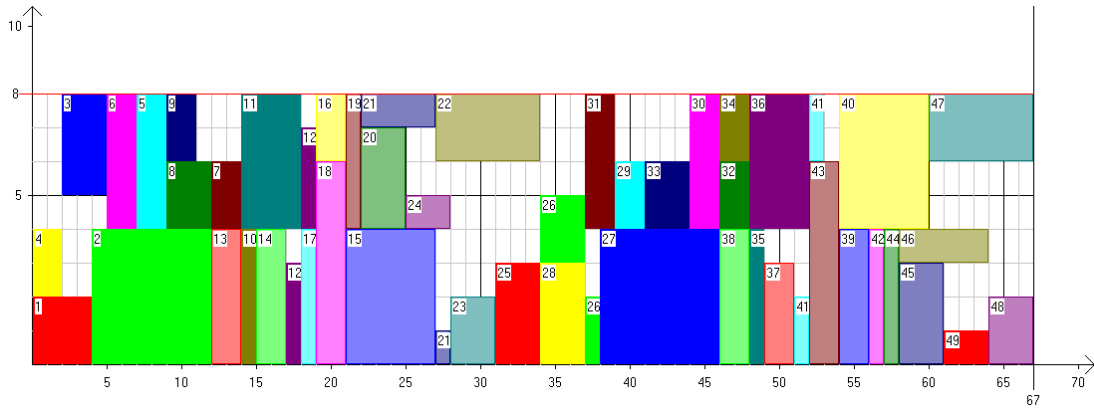


A partir de los resultados obtenidos por el RESCON, el sistema genera la óptima planificación y asignación de recursos posible, tomando como premisa la disminución el los tiempo de desarrollo del proyecto y optimización en el uso de los recursos disponibles.

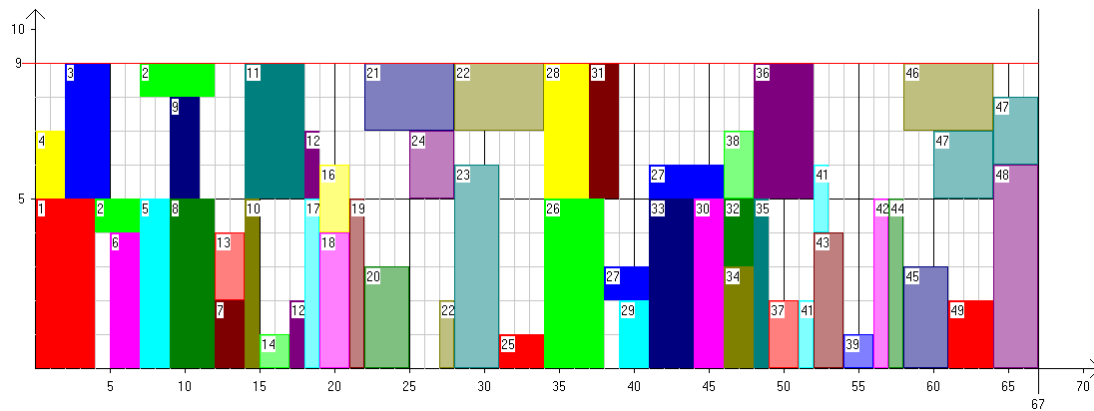
A continuación se presenta un ejemplo de la planificación generada por cada tipo de recurso:



Asignación de tareas: recursos tipo 1

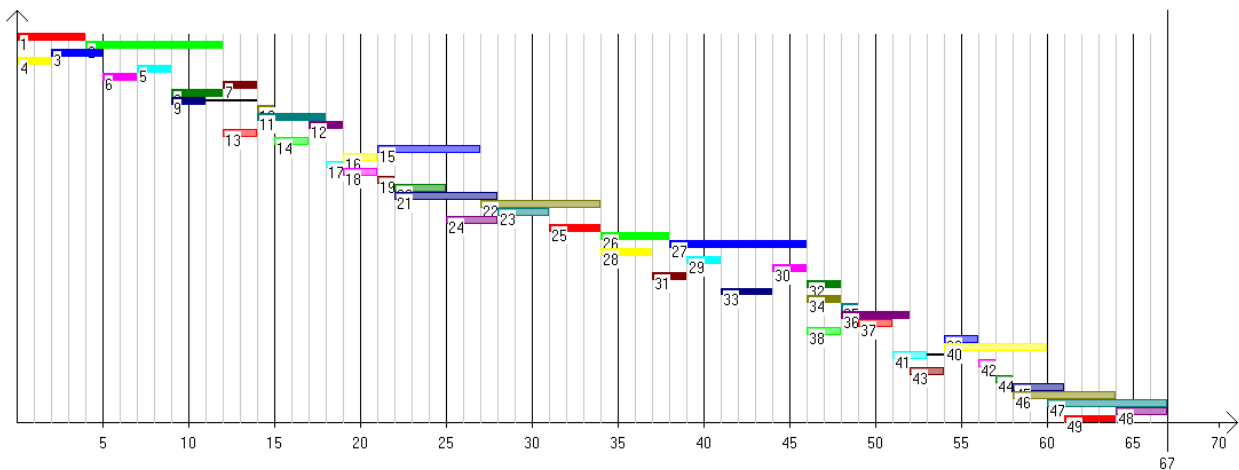


Asignación de tareas: recursos tipo 2



Asignación de tareas: recursos tipo 3

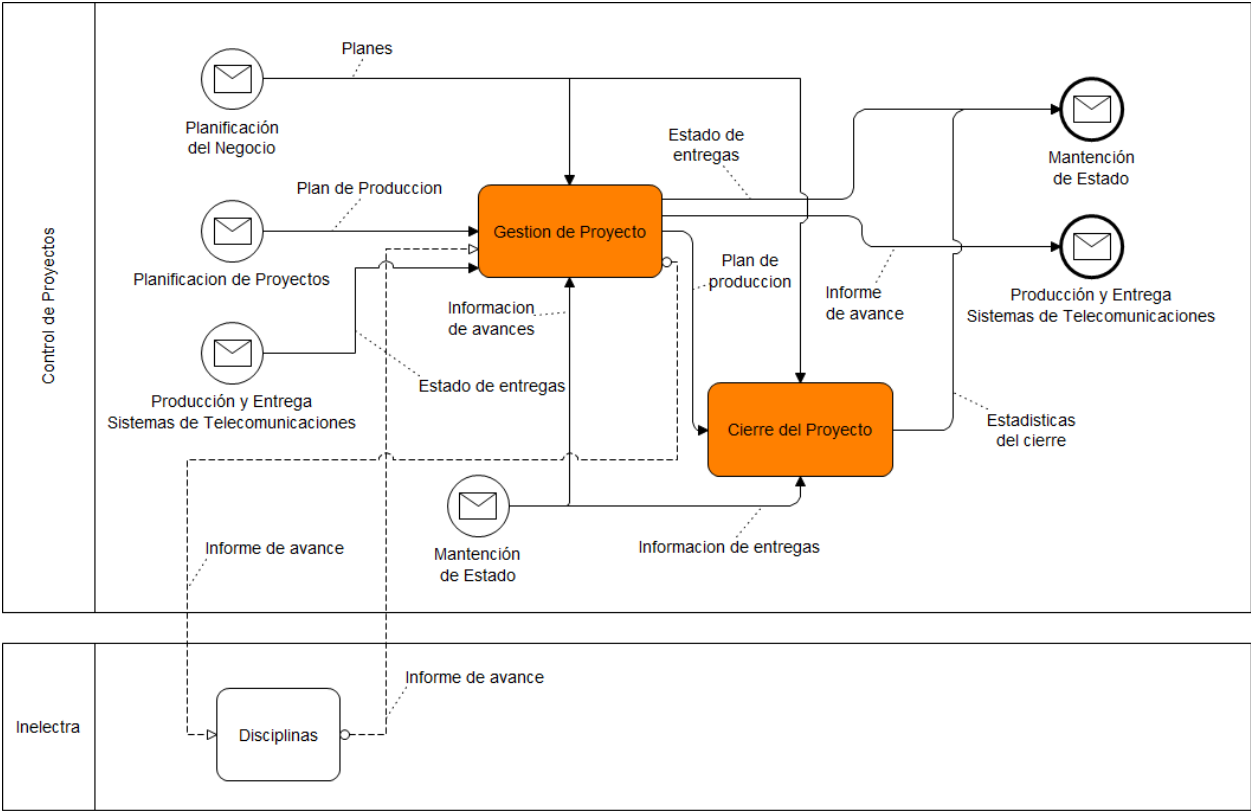
Así mismo el sistema arroja la Carta Gantt para el proyecto definido. A continuación se presenta un ejemplo.



Más adelante se demostrará mediante una prueba de concepto, que los resultados obtenidos mediante el uso de herramientas para solución de problemas de planificación tipo RCPS, mejoran significativamente la asignación de actividades y la planificación de proyectos. Estos resultados se miden en base a la disminución del tiempo de ejecución de los proyectos y la maximización del uso de recursos mediante la óptima asignación de actividades.

### 6.3. Control de Proyectos

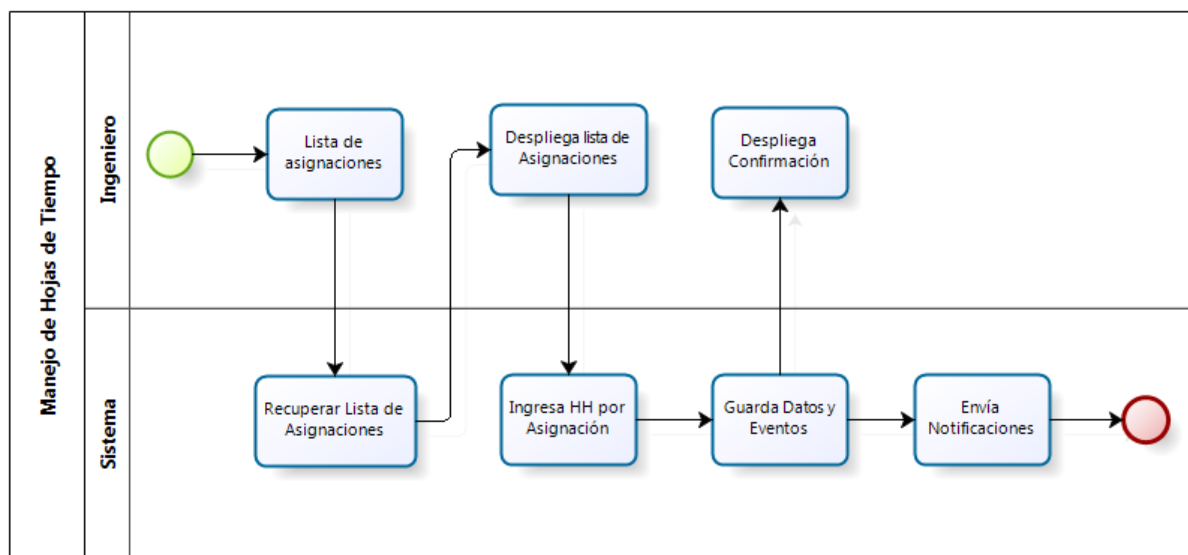
El proceso Control de Proyectos, contempla los subprocesos Gestión de Horas-Hombre y Cierre del Proyecto, los cuales alimentan constantemente el sistema con información sobre los estados de entrega y atrasos de los distintos productos de telecomunicaciones en los diferentes proyectos activos.



El subproceso **Gestión de Proyectos**, contempla la gestión de la planificación de las horas-hombre asignadas a los recursos de un proyecto, específicamente, la

administración de hojas de tiempo, estado de entrega de los productos en los plazos estipulados e información de estado del proyecto por parte de los ingenieros, líderes de telecomunicaciones y otras disciplinas de Inelectra.

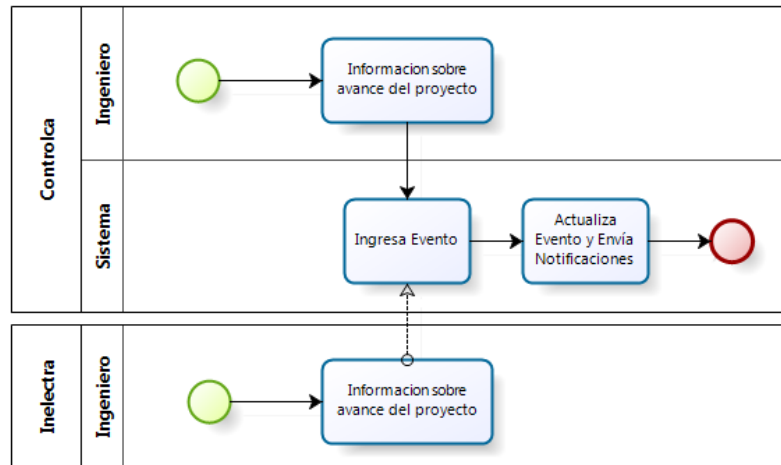
A continuación se detalla en BPMN el subproceso **Manejo de Hojas de Tiempo** (Control de Horas Hombre).



En el subproceso Manejo de Hojas de Tiempo se detalla la interacción entre el usuario y el sistema de gestión de hojas de tiempo, el cual anteriormente, se llevaba a cabo en forma manual mediante hojas de Excel. Al implementar esta solución, se podrá tener acceso, en tiempo real, al estado de avance de los proyectos y de esta forma evitar contratiempos.

A continuación se detalla en BPMN el subproceso **Control de Eventos**





De forma similar que el subproceso anterior, el subproceso Estado de Proyecto permitirá conocer y actualizar el estado del proyecto dado cambios de alcance y avances en entregas, tanto por Controlca como por Inelectra, permitiendo de esta manera su interacción.

## **7. DISEÑO IMPLEMENTADO COMO APOYO A LOS PROCESOS**

El entorno de desarrollo IDE que se empleará es Microsoft Visual Estudio 2010, el cual es un ambiente de desarrollo integrado para sistemas operativos Windows que soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, entre otros.

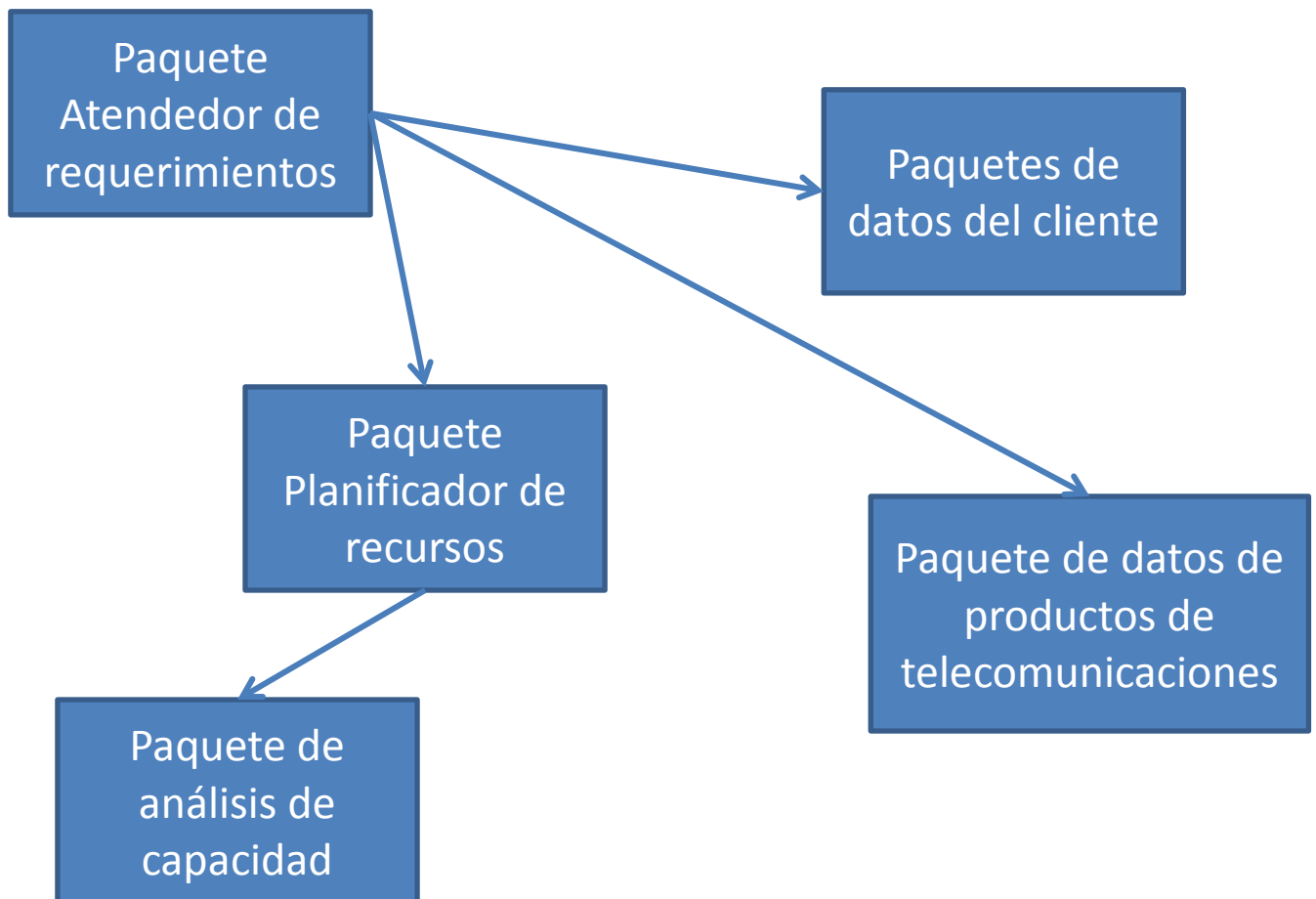
Visual Studio permite crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET.

### **7.1. DIAGRAMA DE PAQUETES**

En resumen, la arquitectura de sistemas de apoyo al proceso de “Planificación Programación y Control Proyectos Telecomunicaciones” se basa en los procesos relacionados a gestión y planificación de proyectos donde los aspectos de lógica relevantes tienen que ver con la evaluación de recursos necesarios para la ejecución del proyecto, la priorización de tales proyectos y la asignación de recursos para

desarrollar el diseño y la evaluación del proyecto. Ésto termina con un plan detallado de trabajo para cada proyecto de telecomunicaciones, incluyendo tareas, personal asignado a cada tarea, fechas en las cuales deben terminar las tareas y, como consecuencia la fecha de finalización del proyecto. Una parte importante de esta lógica es la asignación de las personas a los proyectos, de acuerdo a sus habilidades y experiencia, y la estimación de los tiempos para las diferentes tareas del proyecto, la cual puede basarse en antecedentes estadísticos de proyectos anteriormente realizados.

A continuación se presenta la arquitectura de paquetes que se deriva del análisis del proyecto a realizar.



En el caso particular que se pretende desarrollar, una posible estructura general de sistemas podría incluir: un paquete atendedor de requerimientos, que apoye todo el contacto inicial con el cliente hasta terminar con la definición de los productos de

telecomunicaciones que requieren; un paquete de datos del cliente, que contendría un repositorio de datos de proyectos antiguos y datos relevantes para la creación de ofertas; un paquete de datos de productos de telecomunicaciones que contenga repositorio de posibles productos de telecomunicaciones a ser ofrecidos; un paquete planificador de recursos que posea la lógica de determinación de recursos; y un paquete de análisis de capacidad que posea la lógica de asignación de tareas.

## 7.2. CASOS DE USO

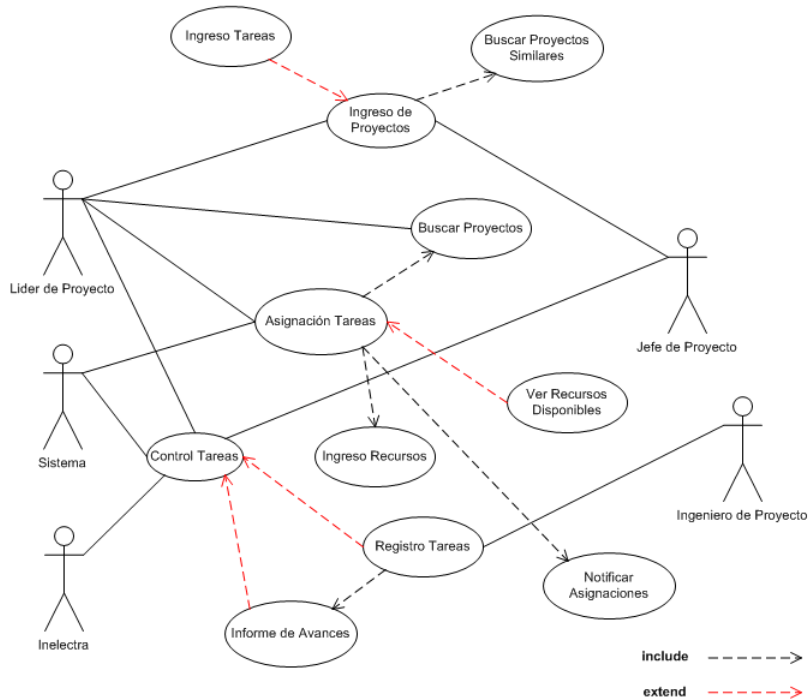
El diagrama de casos de uso considera la presencia de 5 actores relevantes que son: Líder de Proyecto, Jefe de Proyecto, Ingeniero de Proyecto, Inelectra y “Demonio”. A continuación se describen sus roles.

- **Líder de Proyecto:** este rol es desempeñado por el Gerente General, quien maneja toda la cartera de proyectos.
- **Jefe de Proyecto:** ingeniero encargado de la planificación y control del proyecto. Puede desempeñar el rol de “Ingeniero de Proyecto” en el mismo proyecto y /o en otros.
- **Ingeniero de Proyecto:** ingeniero que desarrolla los productos a ser entregados al cliente final.
- **Inelectra:** aliado estratégico, que interviene activamente en el avance del proyecto.
- **Sistema:** actividades desarrolladas por el mismo sistema en forma de tareas programadas, ejecutadas por un “trigger”.

Es importante resaltar que el rol de “Jefe de Proyecto” puede ser desempeñado por el mismo Líder de Proyecto, quien en ese caso sería el único responsable por el proyecto que le sea asignado. A su vez, tanto el Jefe de Proyecto como el Líder de Proyecto pueden desempeñar el rol de “Ingeniero de Proyecto” en cualquier proyecto.

Los Jefes de Proyecto son apoyados directamente por los Ingenieros de Proyecto, quienes pueden desarrollar tareas de distintos proyectos.

Los casos de uso diseñados cubren desde el ingreso de nuevos proyectos hasta el control y seguimiento de tareas. A continuación se muestra el diagrama de casos de uso que muestra las relaciones entre estos y los actores antes mencionados:



A continuación se detallarán cada uno de los casos de uso ilustrados en el diagrama anterior.

### 7.2.1. Detalles de los Casos de Uso

#### 7.2.2. Ingreso Proyectos

Permite el ingreso de nuevos proyectos, indicando para ello: idioma, localización, factor de riesgo, tipo de proyecto, complejidad, entre otros. Posteriormente el sistema procesa la información y genera la lista de productos entregables y el total de horas-hombre estimadas para su desarrollo.

### **7.2.3. Ingreso Tareas**

Durante el ingreso de un nuevo proyecto se podrán ingresar nuevos productos y tareas inexistentes en el sistema, los cuales quedarían disponibles para proyectos en ejecución y futuros proyectos.

### **7.2.4. Buscar Proyectos Similares**

Tiene relación directa con el caso de uso anterior, ya que es mediante esta funcionalidad que es posible obtener las tareas y productos que componen a un nuevo proyecto.

### **7.2.5. Buscar Proyectos**

Permite hacer una búsqueda de proyectos para visualizar los estados de avance y atraso, así como la asignación de nuevas tareas.

### **7.2.6. Asignación Tareas**

Es de gran relevancia ya que constituye el proceso de incorporación de recursos a cada una de las tareas a implementar de manera óptima, es decir, incrementando el uso de recursos de manera de disminuir los tiempos ociosos, aumentando la productividad. También llamado asignación de recursos a tareas, contempla la búsqueda de proyectos y por ende tareas a implementar, junto con la utilización de los recursos disponibles, y el ingreso de recursos nuevos o de apoyo, que serán utilizados en un modelo de asignación que entregue como resultado las asignaciones de los distintos recursos a cada una de las tareas, y que por último se les notifica de dicha asignación por medio de un mensaje por una vía de comunicación como el correo electrónico.

### **7.2.7. Ver Recursos Disponibles**

Permite realizar la búsqueda de recursos disponibles desde el repositorio de recursos, junto con lo cual se determinará inicialmente cuáles son adecuados para realizar las tareas encomendadas, y que por ende, entrega solamente los recursos que estén capacitados. También se considerará su capacitación en caso de ser necesario.

### **7.2.8. Ingreso Recursos**

Forma parte del proceso de asignación, de manera que permite la incorporación de nuevos recursos, comúnmente Ingenieros de Proyecto y que son ingresados al repositorio correspondiente para ser posteriormente incorporados al proceso de asignación.

### **7.2.9. Notificar Asignaciones**

Una vez realizada la asignación, se envía un mensaje destinado a cada recurso asignado, informándole de su tarea a realizar y de todos aquellos aspectos que ésta implica, como el tiempo estimado, requerimientos, consideraciones iniciales, etc., de manera de introducirlo en su nueva tarea.

### **7.2.10. Control de Tareas**

Una vez que se ha realizado la asignación de recursos, y que posteriormente se ha iniciado la ejecución de los proyectos, se ejecuta el control y seguimiento de las tareas. Por un lado se tiene que el responsable se encarga del manejo global de los proyectos corroborando que se cumplan los plazos y requerimientos generales, dando instrucciones a los distintos Ingenieros de Proyecto. El Jefe de Proyectos sigue la implementación de cada una de las tareas de manera tal que cada uno de los ingenieros involucrados reciba las instrucciones de implementación. Finalmente cada ingeniero registra sus avances diarios en el sistema y envía peticiones a su Jefe de Proyectos en caso de requerirlo, a través del caso de uso Registro de Tareas.

### **7.2.11. Registro de Tareas**

Es precisamente el registro de horas-hombre consumidas, incidencias, problemas y soluciones generadas semanalmente durante el desarrollo de una tarea, y que tiene por finalidad controlar el estado de avance o atraso de las tareas del proyecto.

### **7.2.12. Informe de Avances**

Este caso de uso es de suma importancia ya que permite conocer el estado de avance y atraso de cada una de las tareas asociadas al proyecto.

## **7.3. DIAGRAMAS DE ESCENARIOS**

A continuación se muestran los diferentes diagramas de escenarios que describen en forma general a la aplicación de “ingreso de proyectos, planificación, y control de proyectos”. Estos diagramas entregan una visión macro de la interacción entre los actores y el sistema.

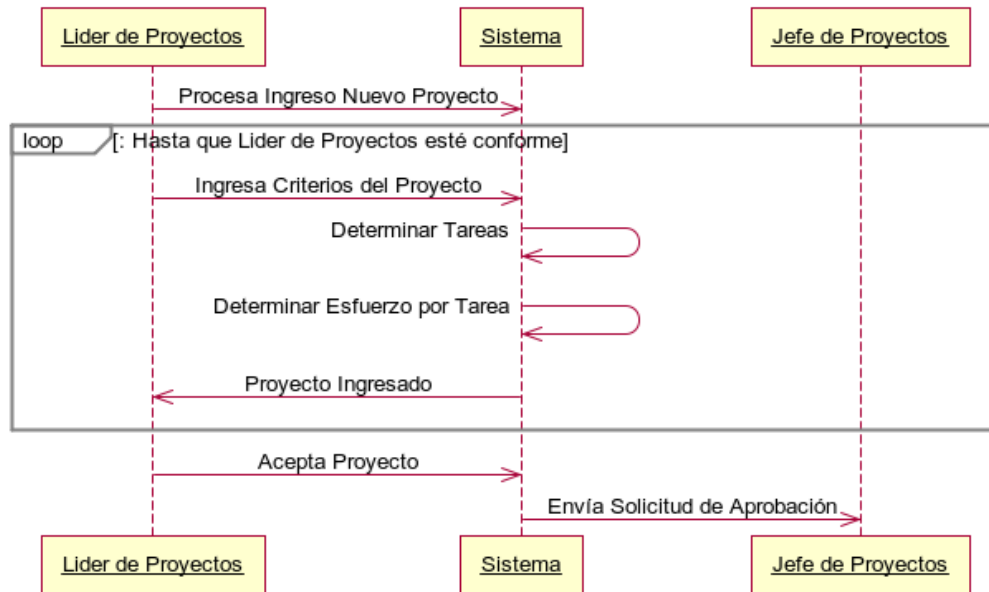
Posteriormente, basándonos en estos diagramas procederemos a desarrollar los diagramas de secuencia extendidos o diagramas de realización, tomando como referencia la literatura del Dr. Oscar Barros.

Los diagramas de escenarios pueden ser divididos en tres grupos, basados en sus funcionalidades.

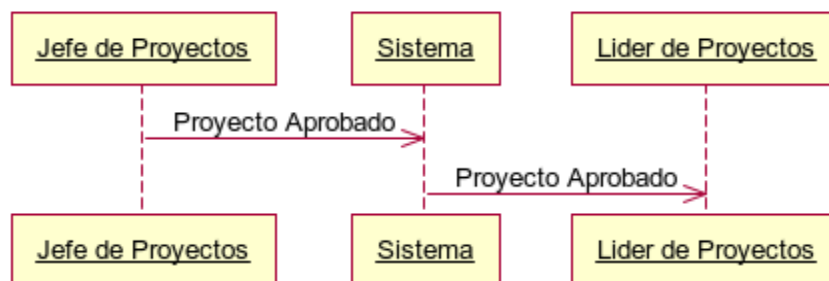
- Ingreso de Proyectos
- Planificación de Proyectos
- Control de Proyectos

### 7.3.1. Ingreso de Proyectos

A continuación se presenta el diagrama de escenario “Ingreso de Proyectos”, que como bien lo indica su nombre, involucra al Líder del Proyecto en la creación de un nuevo proyecto.



El diagrama anterior ilustra en forma general la interacción con el sistema de nuevos proyectos al sistema, de forma tal de permitir su clasificación inmediata de acuerdo a las características y criterios definidos para el proyecto ingresado. Para ello el sistema genera un listado de los productos a ser desarrollados en el proyecto y estima el total de horas-hombre asignadas a cada uno, tomando como referencia la información ingresada por el Líder de Proyectos. Esta información es almacenada en un repositorio de datos para su posterior aprobación.

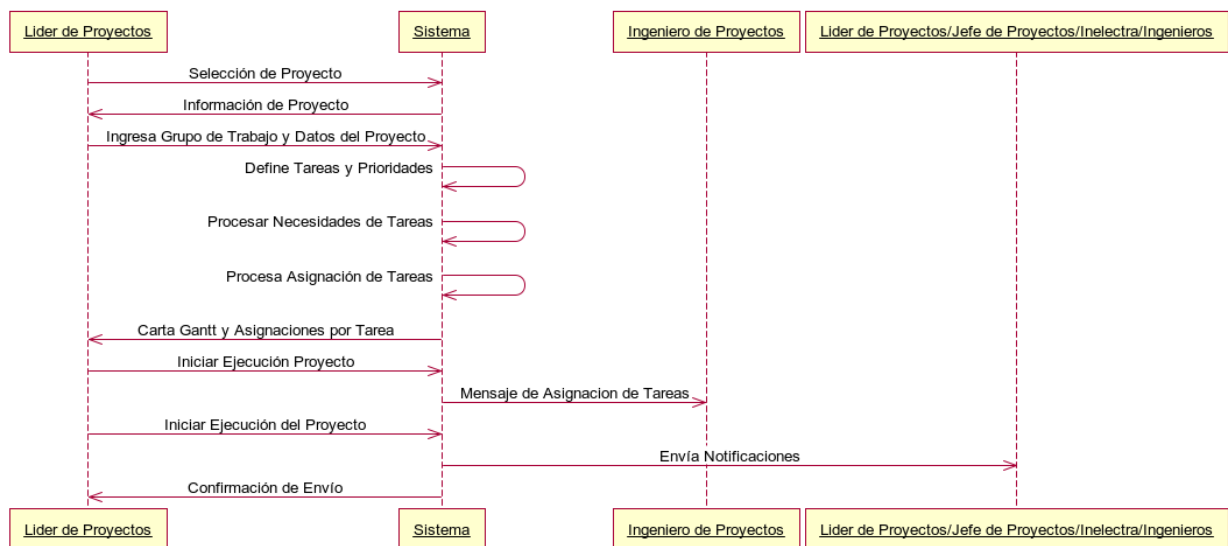




El proceso de ingreso del nuevo proyecto finaliza una vez que el Líder de Proyecto aprueba dicho ingreso.

### 7.3.2. Planificación de Proyectos

El diagrama de escenario “Planificación de Proyectos” detalla el proceso de planificación y entrega de productos, así como la asignación de tareas a los distintos ingenieros involucrados en el proyecto.



El diagrama anterior refleja el proceso de definición de prioridades, asignación de recursos y tareas del Proyecto. Es aquí cuando se ejecuta la lógica de planificación, la cual determina los recursos disponibles, y dado ciertos parámetros de priorización definidos por el usuario y explicados en el punto 4.2, realiza las asignaciones de tareas a los distintos ingenieros capacitados para la ejecución de dichas tareas. Así mismo, define los plazos de entrega y sus prioridades (carta Gantt).

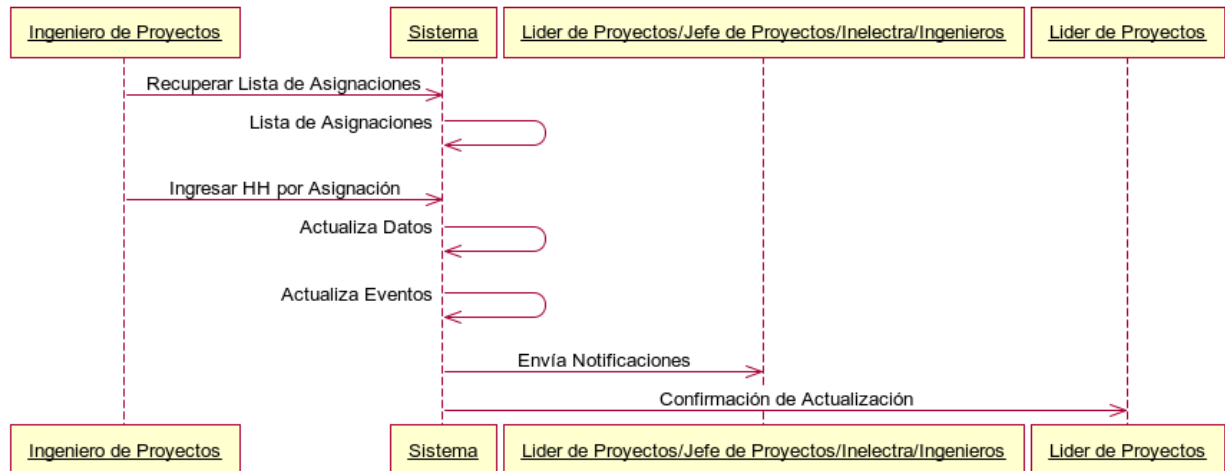
### 7.3.3. Control de Proyectos

El control de proyectos se divide en dos grupos, “Control de Horas Hombre” y “Control de Eventos”. Es importante resaltar que el rol de “Ingeniero de Proyectos” es

común para todos el personal del Proyecto, por lo que todos, incluyendo al Líder y Jefe de Proyecto deben registrar las horas hombre trabajadas en el proyecto.

### Control de Horas Hombre

- Ingeniero de Proyecto



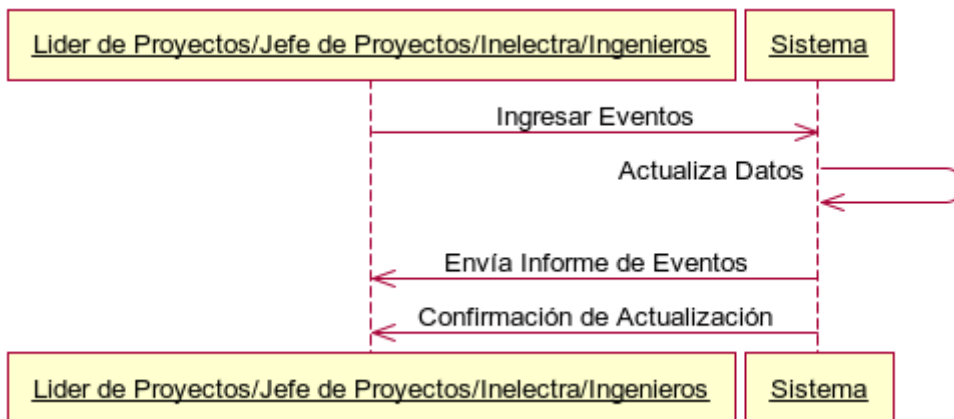
En diagrama anterior refleja la interacción del Ingeniero de Proyectos con el sistema de control de control de proyectos, mediante el “Control de Horas Hombre”, a través del cual cada ingeniero involucrado en el proyecto, ingresa al sistema la cantidad de horas empleadas en el desarrollo de cada producto.

Cualquier cambio generado en el sistema, es informado al Líder de Proyecto, Jefe de Proyecto e instancias de Inelectra, pertinentes para ese proyecto particular.

### Control de Eventos

- Líder de Proyecto
- Jefe de Proyecto
- Ingeniero de Proyecto
- Inelectra

Es importante resaltar que el proceso “Control de Eventos” puede realizado indistintamente por: Lider de Proyectos, Jefe de Proyectos, Ingeniero de Proyectos o Inelectra.



El diagrama anterior contempla el control de eventos del proyecto. Como bien lo refleja el diagrama cualquier cambio generado en el sistema, es informado al Líder de Proyecto, Jefe de Proyecto e instancias de Inelectra pertinentes para ese proyecto particular.

Adicionalmente se contempla la interacción de alguna instancia de Inelectra con el sistema de control de proyectos de Controlca. El sistema le permite a Inelectra registrar cambios en eventos del proyecto, entendiendo “evento” como cualquier cambio de dirección que influya significativamente en el progreso del proyecto. Un ejemplo, de evento, puede ser un cambio de alcance, el cual implica una re-planificación en las tareas y recursos del proyecto.

#### 7.4. DIAGRAMAS DE SECUENCIA EXTENDIDOS Y CLASES

En esta fase se establecen las Clases participantes en los Casos de Uso y se les asigna un tipo dentro de la nomenclatura UML, ya sea “Boundary”, “Control” o “Entity”. La Realización de los Casos de Uso, que es la manera en que las clases interactúan en la ejecución de la lógica, se presenta por medio de Diagramas de Secuencia Extendidos o realizaciones, el cual corresponde al diseño lógico del sistema.

Dicho diseño se realiza identificando los componentes de la arquitectura general de apoyo a los procesos y se mapean a clases UML como sigue:

- Interfaz: **Boundary**
- Controlador Interacción, Lógica de Interfaz, Lógica de Negocio: **Control**
- Capa datos: **Entity**

Con las clases identificadas se construirán Diagramas de Secuencia Extendidos o Realizaciones, que detallan la manera en que las clases interactúan en la ejecución de la lógica, y Diagramas de Clases que detallan las relaciones entre éstas.

Los Diagramas de Realización pueden ser divididos en tres grupos, basados en sus funcionalidades.

- Ingreso de Proyectos
- Planificación de Proyectos
- Control de Proyectos

#### 7.4.1. Ingreso de Proyectos

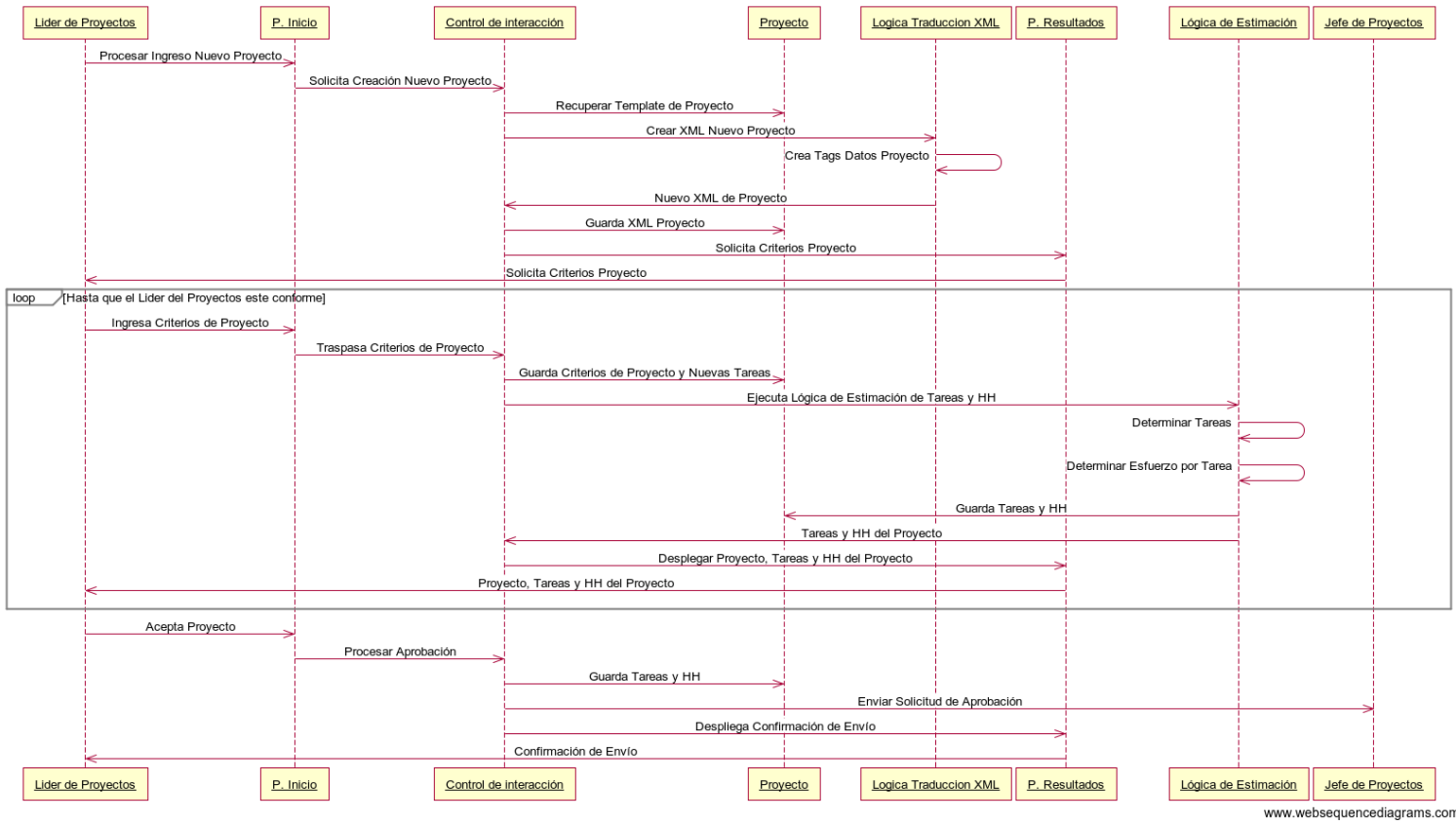
Las clases necesarias para esta realización se muestran a continuación:

- **Boundary**
  - ✓ **Página de inicio:** página web que le permite al usuario interactuar con el sistema. En este caso, permitirle establecer los criterios que definen el proyecto.
  - ✓ **Página de resultados:** página web que permite a la aplicación entregarle los resultados de las consultas realizadas. En este caso, el sistema despliega la lista de productos

- **Control**
  - ✓ **Control de interacción:** funciona como control principal y es a través del cual se administra la comunicación entre los Boundary y los controles ejecutores de lógica.
  - ✓ **Lógica de Traducción XML:** interpreta la información almacenada en formato XML referente a los criterios de selección del proyecto y genera una página web en html.
  - ✓ **Lógica de estimación:** determina en base a los criterios definidos por el usuario los productos e telecomunicaciones a ser realizados en un proyecto específico, así como su valor de horas hombre.
  
- **Entity**
  - ✓ **Proyecto:** clase cuyos atributos definen los datos que se almacenarán en una base de datos relativa al proyecto; éstos se registran en esta base de datos mediante los controles de interacción, los cuales a su vez toman información almacenada que define los criterios de selección del proyecto.

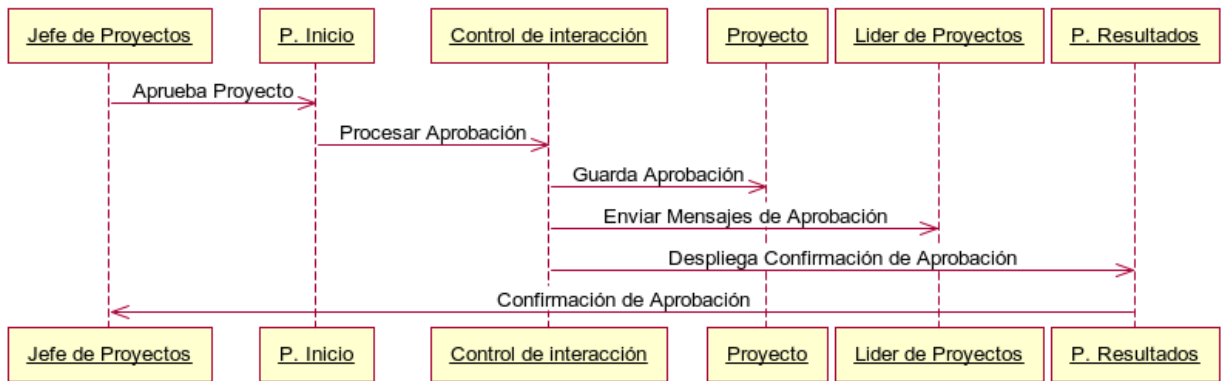
El diagrama de realización se divide en dos bloques, ya que aunque son situaciones que ocurren en forma secuencial, son iniciados por actores distintos en momentos distintos.

A continuación se presenta el diagrama de realización: "Ingreso de Proyecto".



www.websequencediagrams.com

Seguidamente se presenta el diagrama de realización: “Aprobación de Proyecto”.



www.websequencediagrams.com

El diagrama “Ingreso de Proyecto” detalla el ingreso de cada nuevo proyecto al sistema. El ingreso de proyectos es una tarea exclusiva del Líder de Proyectos, quien dada la información proporcionada por Inelectra, procede a definir los criterios de creación de un proyecto, seleccionando entre las opciones disponibles, aquellas que definen de mejor manera las condiciones indicadas por el cliente.

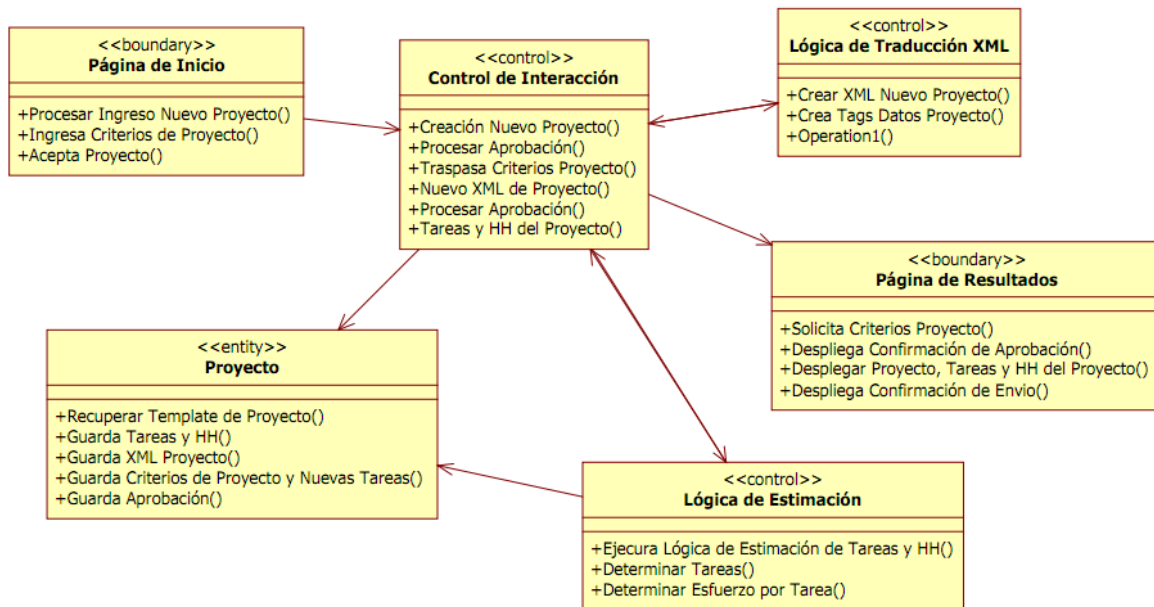
El proceso de definición de tareas y ponderación de horas hombre lo realiza el sistema en base a los criterios definidos por el Líder de Proyectos. Esta tarea la realiza el control “Lógica de Estimación” en base a una lista base de productos predefinidos, los cuales tienen un valor de horas hombre mínimo, el cual se va incrementando dada la información suministrada por el Líder de Proyectos.

En base a la historia de proyectos de Controlca, se ha construido una lista base de todos los posibles productos que podrían ofertarse. Para cada uno de ellos se ha definido un valor mínimo de horas hombres necesarios para su realización, tomando como referencia las condiciones menos exigentes de trabajo.

El sistema, a través de la clase “Lógica de Estimación” determina en base a las condiciones del proyecto los productos que deberán ofertarse y para cada uno su valor de horas-hombre. Esta clase a través de sus métodos analiza cada producto y ajusta su valor de horas-hombre dados los criterios definidos.

El resto del proceso queda explícito en los diagramas, ya que no requieren de una lógica compleja, sino más bien una lógica simple de interacción, entre el sistema y los actores.

A continuación se presenta el diagrama de clases referente al proceso “Ingreso de Proyectos”, el cual refleja la interacción entre las distintas clases que forman el sistema.



## 7.4.2. Planificación de Proyectos

Las clases necesarias para esta realización se muestran a continuación:

- **Boundary**
  - ✓ **Página de inicio:** permite definir el grupo de trabajo del proyecto.
  - ✓ **Página de resultados:** el sistema despliega la carta Gantt y asignaciones por tarea.
  
- **Control**
  - ✓ **Control de interacción:** funciona como control principal y es a través del cual se administra la comunicación entre los Boundary y los controles ejecutores de lógica.
  - ✓ **Lógica de Priorización:** genera la planificación del proyecto en base a la lista de tareas asignadas al proyecto.
  - ✓ **Lógica de Staffing Multiskill:** asigna las tareas a los ingenieros del proyecto en base a información de disponibilidad y nivel de conocimiento de los mismos.

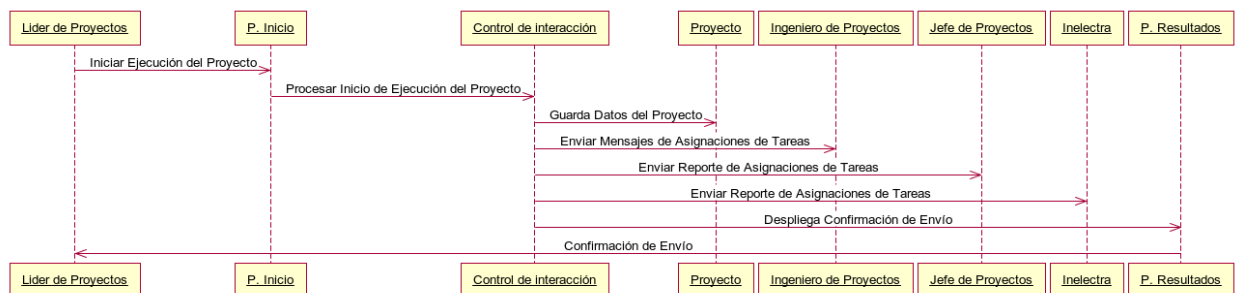
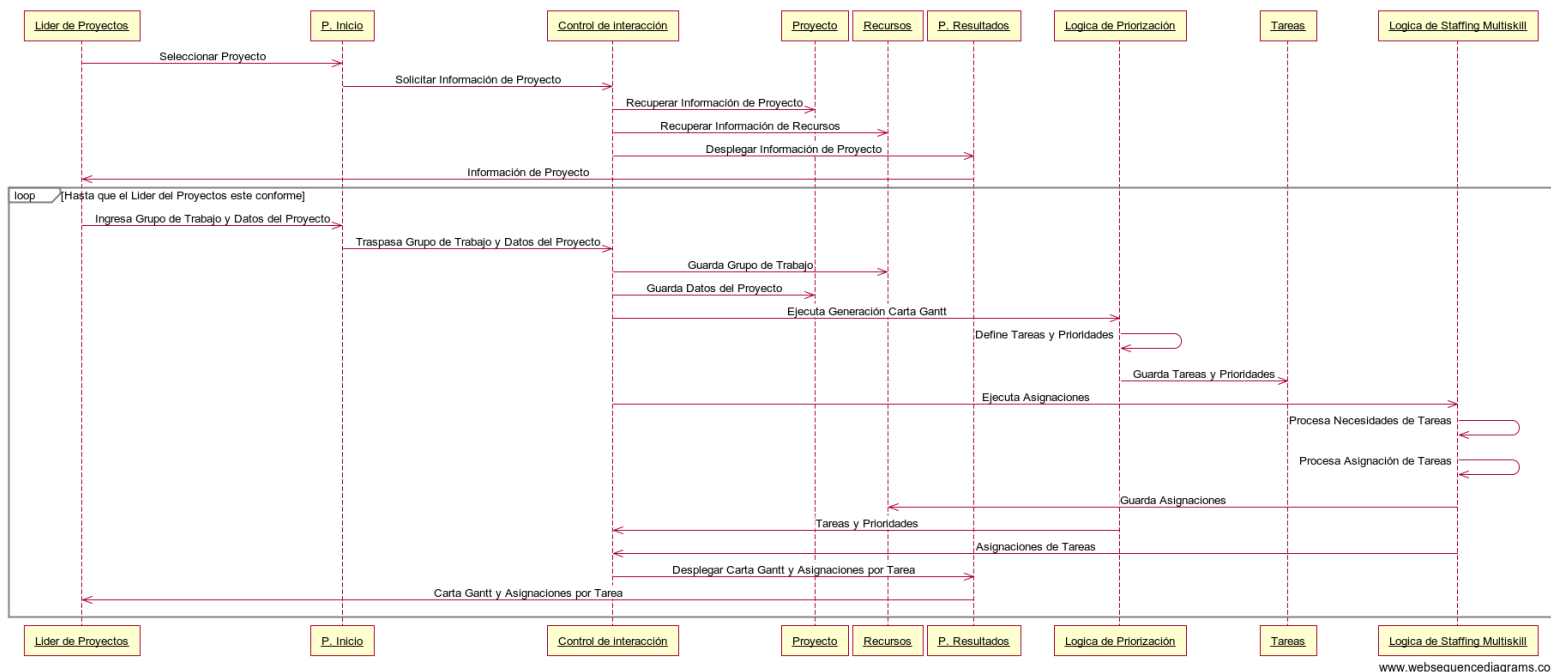


- Entity

- ✓ **Tarea:** define los datos relacionados a las asignaciones del proyecto y su avance
- ✓ **Proyecto:** clase cuyos atributos definen los datos que se almacenarán en una base de datos relativa al proyecto
- ✓ **Recursos:** define los datos relacionados al personal de la empresa, así como sus conocimientos y destrezas

El diagrama de realización se divide en dos bloques, dada su longitud.

A continuación se presentan los diagramas de realización: “Planificación de Proyectos”.

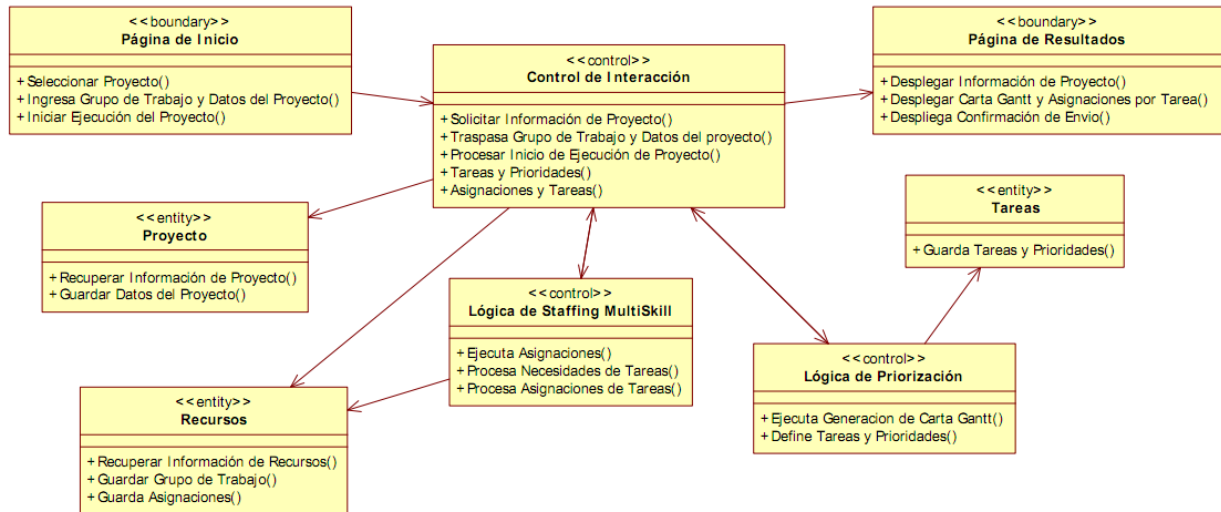


Los diagramas presentados anteriormente detallan el proceso de planificación de cada nuevo proyecto ingresado al sistema. Esta tarea es realizada por el Líder de Proyectos, quien en base al personal disponible y con el perfil adecuado define su grupo de trabajo para el desarrollo del proyecto.

Los dos puntos más relevantes son los relacionados con las lógicas de priorización y asignación de recursos. La lógica de priorización toma como referencia información preestablecida sobre reglas de prioridad, es decir, cada producto tiene asociada una serie de condiciones que le permiten al sistema definir el momento en el deberá planificarse el inicio de dicha actividad. Por otro lado, la lógica de asignación de recursos se basa en el perfil de destrezas y conocimientos de cada ingeniero, así como su disponibilidad horaria para definir el grupo de ingenieros que formará parte del grupo de trabajo del proyecto.

Para que el sistema, a través de la lógica de asignación de recursos al proyecto defina el grupo de trabajo, deberá ingresarse al sistema información sobre el nivel de conocimiento de cada ingeniero en las distintas aéreas de telecomunicaciones sobre las cuales trabaja Controlca.

A continuación se presenta el diagrama de clases referente al proceso “Planificación de Proyectos”, el cual refleja la interacción entre las distintas clases que forman el sistema.



### 7.4.3. Control de Proyectos

El control de proyectos se divide en dos grupos, “Control de Horas Hombre” y “Control de Eventos”. Ambos diagramas de de realización comparten las mismas clases, pero su funcionamiento es diferente.

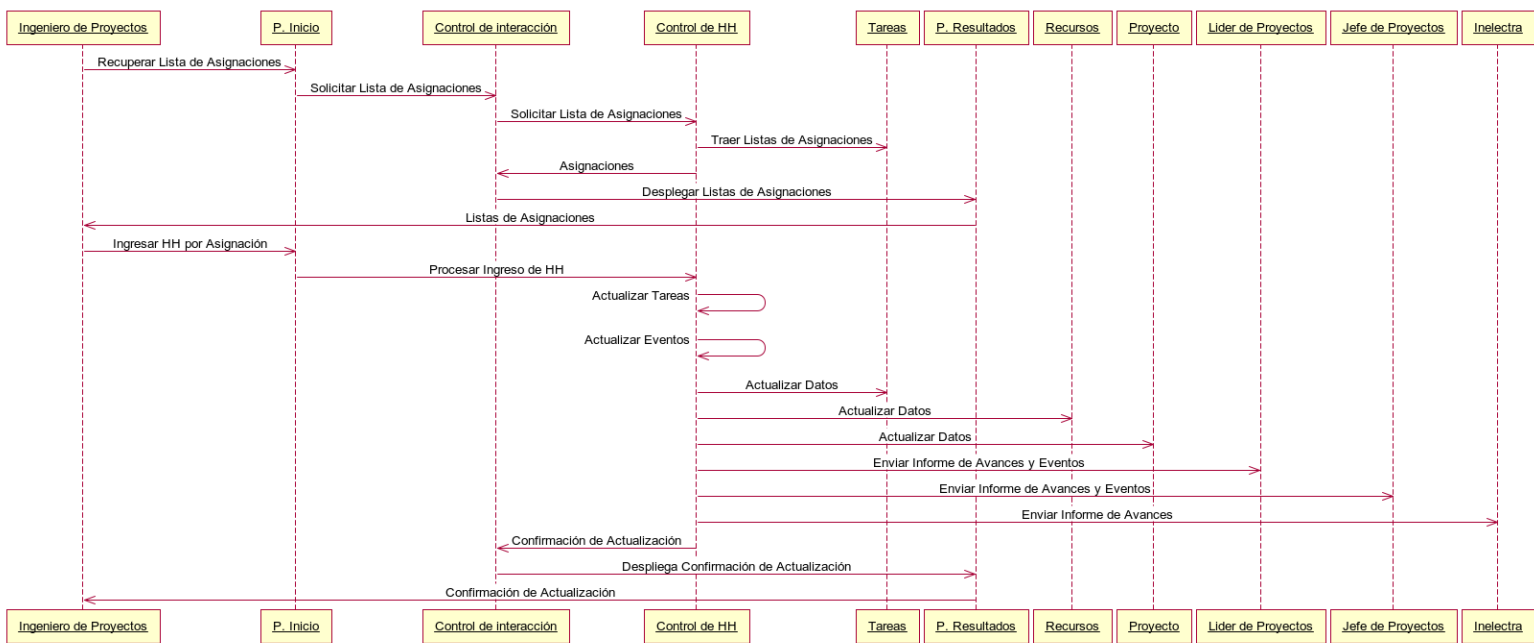
Las clases necesarias para esta realización se muestran a continuación:

- **Boundary**
  - ✓ **Página de inicio:** permite ingresar cambios al proyecto a través de eventos, así como las horas hombre empleadas para el desarrollo de un producto.
  - ✓ **Página de resultados:** el sistema despliega información de confirmación sobre avances y eventos.
- **Control**
  - ✓ **Control de interacción:** funciona como control principal y es a través del cual se administra la comunicación entre los Boundary y los controles ejecutores de lógica.
  - ✓ **Control de HH:** actualiza el avance del proyecto en función de eventos y finalización de tareas.

- **Entity**

- ✓ **Tarea:** define los datos relacionados a las asignaciones del proyecto y su avance
- ✓ **Proyecto:** clase cuyos atributos definen los datos que se almacenarán en una base de datos relativa al proyecto
- ✓ **Recursos:** define los datos relacionados

A continuación se presenta el diagrama de realización: “Control de Horas Hombre”.

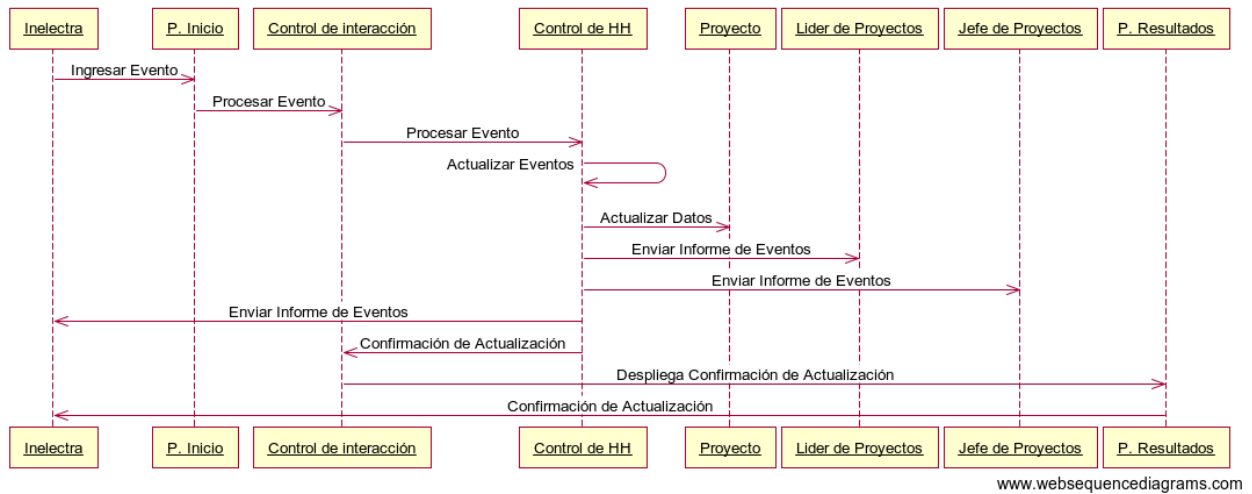


www.websequencediagrams.com

El diagrama presentado refleja la interacción del usuario con el sistema de control de horas hombre, a través del cual se ingresa el valor en horas de los avances de cada producto disponible para dicho usuario. La lógica principal se encuentra en la clase “Control de HH” la cual controla en avance de las tareas del proyecto.

A continuación se presenta el diagrama de realización: “Control de Eventos”, el cual puede ser iniciado por cualquiera de los siguientes actores: Líder de Proyectos, Jefe de Proyectos, Ingeniero de Proyectos e Inelectra.

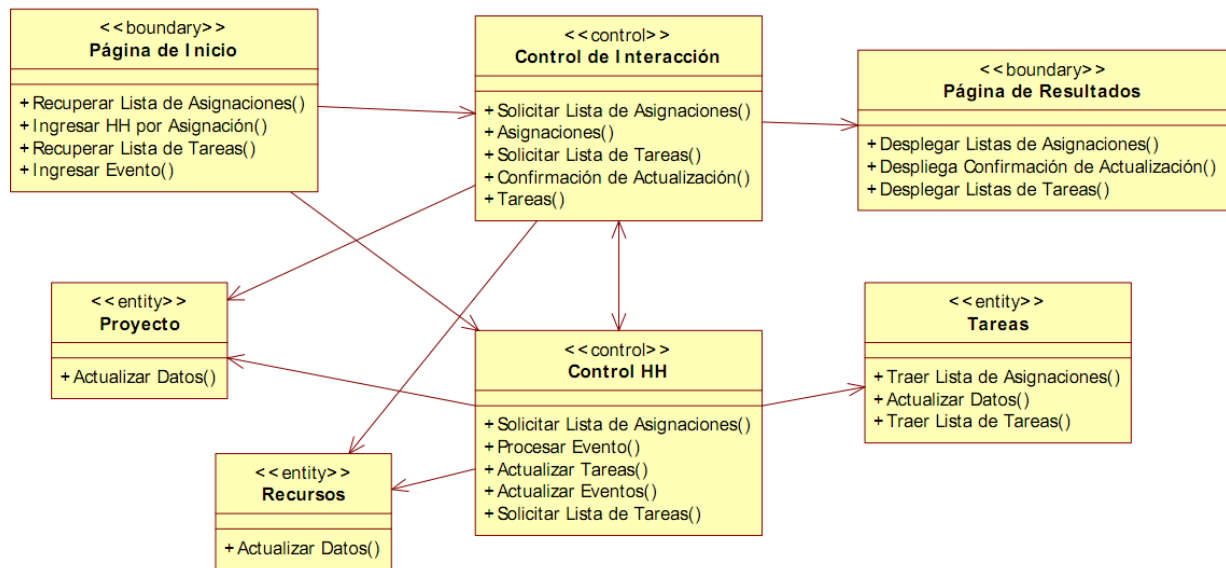
En el diagrama se presentara a Inelectra como iniciador del proceso de ingreso de eventos.



Tal como se indicó anteriormente, a través del sistema de control de proyectos, es posible ingresar eventos que definen el estado de avance del proyecto, dichos eventos son notificaciones sobre el proyecto que sirven para el control del mismo.

Adicionalmente, como ya se comento anteriormente, alguna instancia de Inelectra puede interactuar con el sistema de control de proyectos de Controlca, permitiéndole registrar cambios en eventos del proyecto, entendiendo “evento” como cualquier cambio de dirección que influya significativamente en el progreso del proyecto. Un ejemplo, de evento, puede ser un cambio de alcance, el cual implica una re-planificación en las tareas y recursos del proyecto.

A continuación se presenta el diagrama de clases referente al proceso “Control de Proyectos”, el cual refleja la interacción entre las distintas clases que formar el sistema.



## 8. PRUEBA DE CONCEPTO

Este capítulo demuestra con datos empíricos que el proceso de planificación de proyectos, detallado en el capítulo 6, realiza una planificación y asignación óptima de los recursos disponibles, mejorando hasta en un 20% los resultados obtenidos mediante el esquema actual. Recordemos que el esquema actual está basado en criterio experto, sin mayor apoyo computacional.

La prueba de concepto se realizó con dos proyectos del tipo ingeniería básica. Para cada uno de éstos, se tomaron las siguientes condiciones iniciales: número de actividades, tiempo de cada actividad, orden de prioridad de las actividades, grupo de trabajo definido y perfil de cada recurso.

A continuación se detallan cada uno de los proyectos empleados y sus resultados.

### **Proyecto 1: El Furril Crudo y Agua**

Este proyecto contempla sólo actividades entregables, es decir, no se han tomado en cuenta reuniones de coordinación interdisciplinaria, ni reuniones con los clientes.

Este proyecto requiere ingenieros con las siguientes capacidades:

- Vigilancia electrónica
  - ✓ CCTV
  - ✓ VOCEO
  - ✓ Control de acceso
  - ✓ Detección de intrusos
- Networking
  - ✓ Cableado estructurado
  - ✓ Canalizaciones
  - ✓ LAN
  - ✓ Protección industrial

La capacidad se mide en función de la experticia y se identifica con un valor del 1 al 5, tal como se detalló en el capítulo 6.

A continuación se presenta la equivalencia:

- Vigilancia electrónica: tipo 2
- Networking: tipo 3

A continuación se encuentra el detalle de los ingenieros asignados y su perfil (capacidad):

- Andrés Caraballo: 2
- Gustavo Angulo: 3
- Aurora Armesto: 3
- Luis Eduardo Serpa: 2
- Alexander Herriques: 2

A continuación se presenta la lista de actividades entregables separadas en documentos y planos, indicando en cada caso, la cantidad de horas-hombre requerida, el tipo de actividad y su prioridad.

<b>DOCUMENTOS</b>			
<b>ACTIVIDAD</b>	<b>HH</b>	<b>TIPO</b>	<b>PRIORIDAD</b>
BASES Y CRITERIOS DE DISEÑO SISTEMAS DE TELECOMUNICACIONES	64	3	1
BASES Y CRITERIOS DE DISEÑO SISTEMAS DE PROTECCIÓN INDUSTRIAL	64	3	1
ESPECIFICACIONES GENERALES CABLEADO ESTRUCTURADO	64	3	2
ESPECIFICACIONES GENERALES LAN	64	3	2
ESPECIFICACIONES GENERALES CCTV	64	2	2
ESPECIFICACIONES GENERALES VOCEO	64	2	2
ESPECIFICACIONES GENERALES CONTROL DE ACCESO	64	2	2
ESPECIFICACIONES GENERALES DETECCIÓN DE INTRUSOS	64	2	2
ESPECIFICACIONES GENERALES CANALIZACIONES PARA TELECOMUNICACIONES	64	3	2

<b>PLANOS</b>			
<b>ACTIVIDAD</b>	<b>HH</b>	<b>TIPO</b>	<b>PRIORIDAD</b>
BASES Y CRITERIOS DE DISEÑO SISTEMAS DE TELECOMUNICACIONES	64	3	1
ARQUITECTURA SISTEMAS DE PROTECCIÓN INDUSTRIAL	96	3	1
DIAGRAMA DE BLOQUE DE LA RED LAN	68	3	2
UNIFILAR CABLEADO ESTRUCTURADO	136	3	2
UNIFILAR VOCEO	68	2	2
UNIFILAR CCTV	68	2	2
UNIFILAR CONTROL DE ACCESO	68	2	2
UNIFILAR DETECCIÓN DE INTRUSOS	68	2	2
SIMBOLOGÍA SISTEMAS DE TELECOMUNICACIONES Y PROTECCIÓN INDUSTRIAL	52	3	1
UBICACIÓN DISPOSITIVOS SISTEMAS PROTECCIÓN INDUSTRIAL	36	3	3
UBICACIÓN EQUIPOS Y DISPOSITIVOS SISTEMAS TELECOMUNICACIONES	36	3	3
UBICACIÓN DISPOSITIVOS SISTEMAS PROTECCIÓN INDUSTRIAL	64	3	3
UBICACIÓN EQUIPOS Y DISPOSITIVOS SISTEMAS TELECOMUNICACIONES	64	3	2

El total de horas-hombre requeridas en este proyecto es de 1.496. La prioridad indica el orden en el que deben ejecutarse las actividades, donde 1 es la máxima prioridad.

Este proyecto real se ejecutó en 105 días hábiles requiriendo exclusivamente un equipo de trabajo compuesto por cinco ingenieros con perfiles tipo 2 y tipo 3.

Para realizar la prueba de concepto, se tomaron los datos del proyecto y se ingresaron en el RESCON en formato PSPLIB. Dicha equivalencia se representa de la siguiente manera.

<b>DOCUMENTOS</b>			
<b>HH</b>	<b>TIPO</b>	<b>SEC</b>	<b>ID</b>
64	R2	1	1
64	R2	1	2
64	R2	2	3

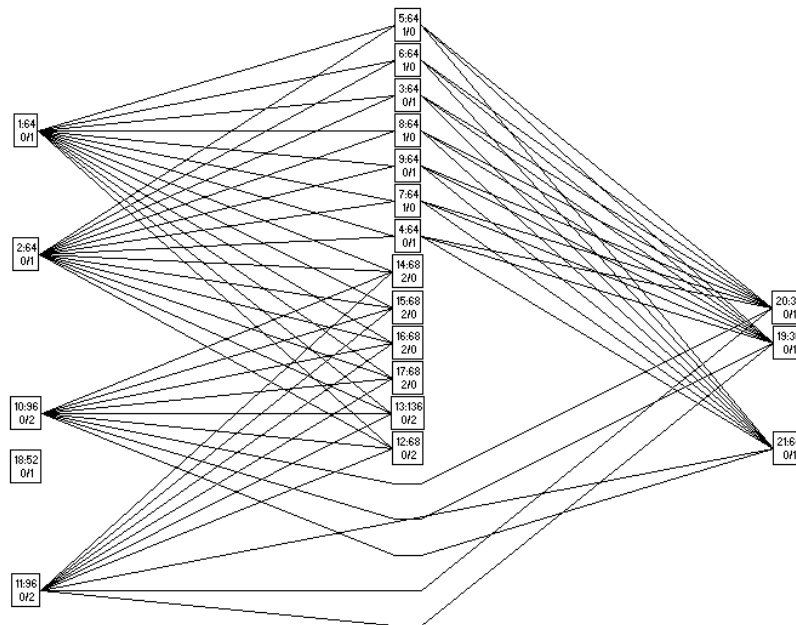
<b>PLANOS</b>			
<b>HH</b>	<b>TIPO</b>	<b>SEC</b>	<b>ID</b>
64	R2	1	10
96	R2	1	11
68	R2	2	12



DOCUMENTOS			
64	R2	2	4
64	R1	2	5
64	R1	2	6
64	R1	2	7
64	R1	2	8
64	R2	2	9

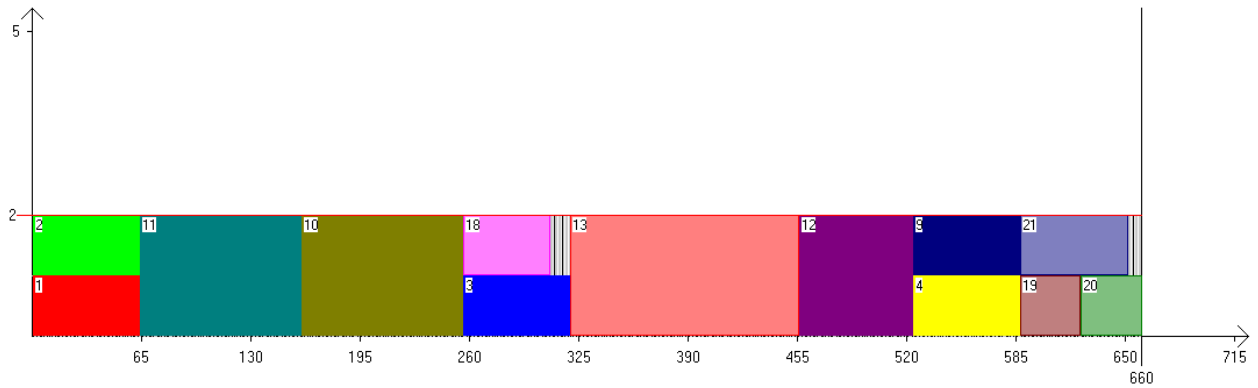
PLANOS			
136	R2	2	13
68	R1	2	14
68	R1	2	15
68	R1	2	16
68	R1	2	17
52	R2	1	18
36	R2	3	19
36	R2	3	20
64	R2	3	21
64	R2	2	22

Al tomar los datos del proyecto e ingresarlos al sistema RESCON obtenemos el siguiente diagrama, que representa las actividades, su relación con otras actividades y los recursos asociados a cada una.

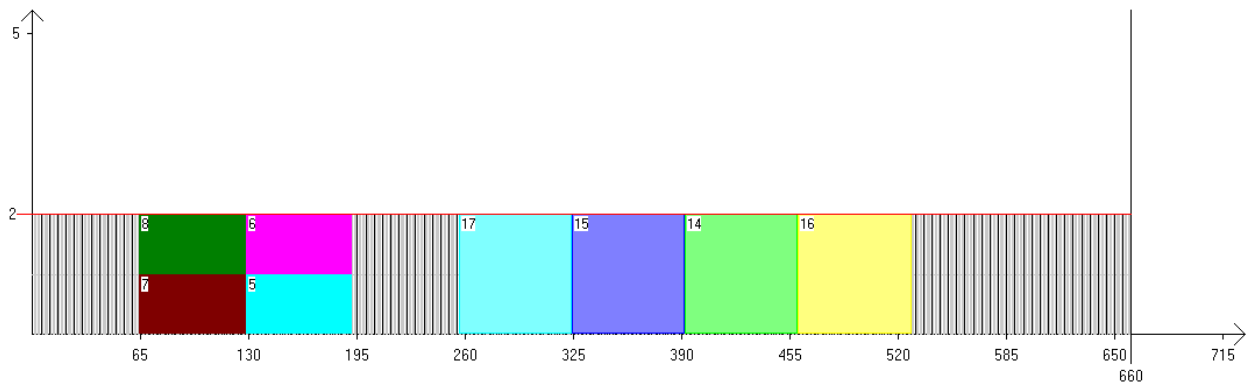


El RESCON emplea una serie de algoritmos de optimización orientados a resolver problemas RCPSP, entre ellos, está el *Branch & Bound* desarrollado por Demeulemeester & Herroelen. Este último será el que usaremos para determinar una solución óptima de asignación de tareas.

A continuación se presenta la óptima asignación de recursos tipo 2 y tipo 3, según la lógica del algoritmo Demeulemeester & Herroelen.



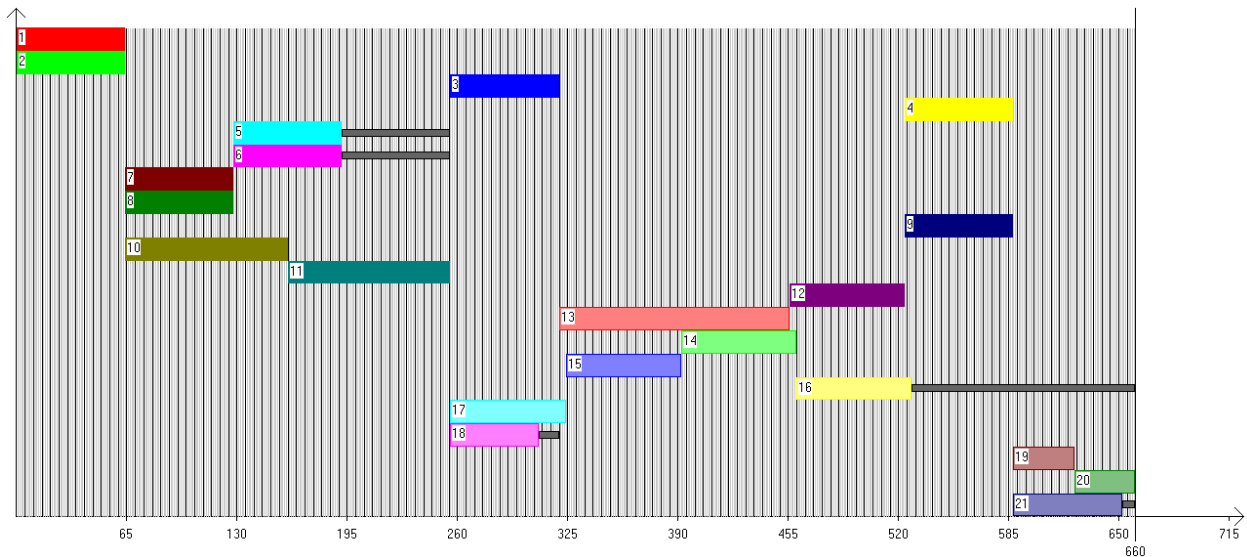
Asignación de recursos tipo 2



Asignación de recursos tipo 3

El eje de vertical representa la cantidad de recursos necesarios, mientras que el horizontal representa la ubicación temporal de la actividad a ser realizada por uno o más recursos en forma paralela. Los espacios sombreados en color gris indican “tiempo ocioso”, es decir, tiempo libre en el que el recurso no tiene tareas disponibles.

De forma similar se obtiene la Carta Gantt del proyecto en base a las asignaciones realizadas por el RESCON, la cual se presenta a continuación.



Carta Gantt

Los resultados obtenidos determinan que el proyecto pudo haberse ejecutado en un plazo máximo de 660 horas hombre, equivalente a 83 días hábiles aproximadamente. Este resultado genera un ahorro de 22 días hábiles, equivalente a un 21% de ahorro del tiempo empleado en el proyecto real.

No obstante, también es posible apreciar que un ingeniero tipo 2 no hubiese sido necesario en el proyecto, lo cual se traduce en una pérdida de aproximadamente 595 horas. Así mismo, dos de los tres recursos tipo 2, quedan libres transcurridas 520 horas desde el inicio del proyecto, lo cual permitiría trasladarlos a nuevos proyectos que requieran expertos con las capacidades tipo 3.

A continuación se presenta una tabla comparativa entre el proyecto real y el proyecto optimizado.

Días usados Real	HH usados Simulación	Ahorro (días)
105	83	22

	Ing. Requeridos Real	Ing. Requeridos Simulación	Ahorro (personal)
Tipo 2	3	2	1
Tipo 3	2	2	0

## Proyecto 2: Siderúrgica Nacional - Infraestructura General

Este proyecto requirió de ingenieros con las siguientes capacidades:

- Vigilancia electrónica: tipo 2
- Networking: tipo 3
- Radiocomunicaciones: tipo 4

A continuación se presenta el detalle de los ingenieros asignados y su perfil (capacidades):

- Ana Lúgia Parra: 2
- Luis Eduardo Serpa: 2
- Juan Marco Rodriguez: 3
- Aurora Armesto: 3
- Patricia Graffe: 4

A continuación se presenta la lista de actividades entregables, indicando en cada caso la cantidad de horas-hombre requerida, el tipo de actividad y su prioridad.

ACTIVIDADES	HH	TIPO	PRIORIDAD
CRITERIOS DE DISEÑO PARA TELECOMUNICACIONES	72	2	1
CRITERIOS DE DISEÑO PARA SISTEMAS DE SEGURIDAD Y VIGILANCIA ELECTRONICA	72	1	1
ESPECIFICACIONES CABLEADO ESTRUCTURADO	60	2	2
ESPECIFICACIONES LAN	60	2	2
ESPECIFICACIONES VOCEO E INTERCOMUNICACION	60	1	2
ESPECIFICACIONES CCTV	60	1	2
ESPECIFICACIONES CONTROL DE ACCESO	60	1	3
ESPECIFICACIONES VIGILANCIA PERIMETRAL	60	2	3
ESPECIFICACIONES SISTEMA DE ENTRETENIMIENTO Y SONORIZACIÓN	72	2	3
ESPECIFICACIONES WAN	60	2	2
ESPECIFICACIONES RADIOCOMUNICACIONES OPERATIVAS	60	3	4
ESPECIFICACIONES WIRELESS LAN	60	2	2
ESPECIFICACIONES GPS	24	3	4
CALCULOS LAN	64	2	4
CALCULOS RADIOCOMUNICACIONES OPERATIVAS	52	3	4
COMPUTOS MÉTRICOS EQUIPOS Y MATERIALES CABLEADO	36	2	4

<b>ACTIVIDADES</b>	<b>HH</b>	<b>TIPO</b>	<b>PRIORIDAD</b>
ESTRUCTURADO (FIBRA ÓPTICA AREA EXTERNA)			
COMPUTOS MÉTRICOS EQUIPOS Y MATERIALES LAN	52	2	4
COMPUTOS MÉTRICOS EQUIPOS Y MATERIALES VOCEO E INTERCOMUNICACION (AREA EXTERNA)	36	1	4
COMPUTOS MÉTRICOS EQUIPOS Y MATERIALES CCTV (AREA EXTERNA)	36	1	3
COMPUTOS MÉTRICOS EQUIPOS Y MATERIALES CONTROL DE ACCESO (AREA EXTERNA)	36	1	2
COMPUTOS MÉTRICOS EQUIPOS Y MATERIALES VIGILANCIA PERIMETRAL	24	1	2

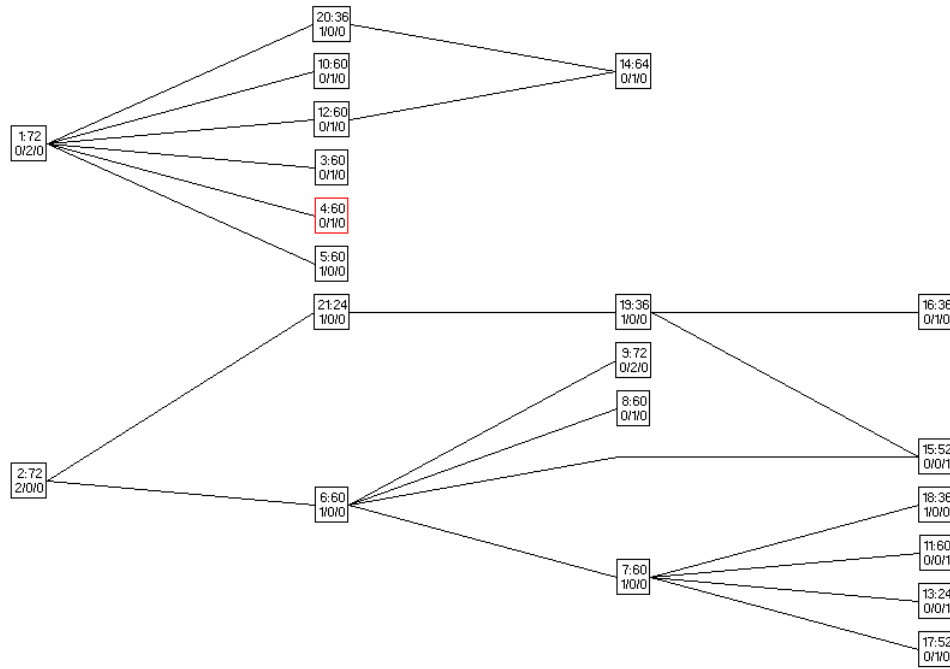
El total de horas-hombre requerido en este proyecto es de 1.116. Este proyecto se ejecutó en 57 días hábiles utilizando exclusivamente un equipo de trabajo compuesto por cinco ingenieros tipo 2, tipo 3 y tipo 4.

A continuación se presenta el proyecto en formato RESCON.

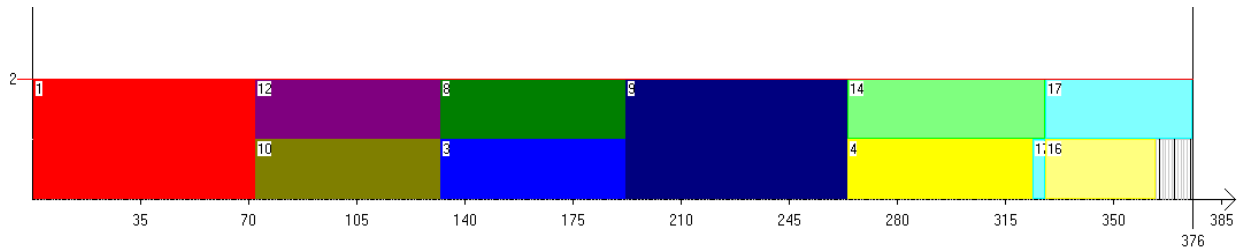
HH	TIPO	SEC	ID
72	R2	1	1
72	R1	1	2
60	R2	2	3
60	R2	2	4
60	R1	2	5
60	R1	2	6
60	R1	3	7
60	R2	3	8
72	R2	3	9
60	R2	2	10
60	R3	4	11

HH	TIPO	SEC	ID
60	R2	2	12
24	R3	4	13
64	R2	4	14
52	R3	4	15
36	R2	4	16
52	R2	4	17
36	R1	4	18
36	R1	3	19
36	R1	2	20
24	R1	2	21

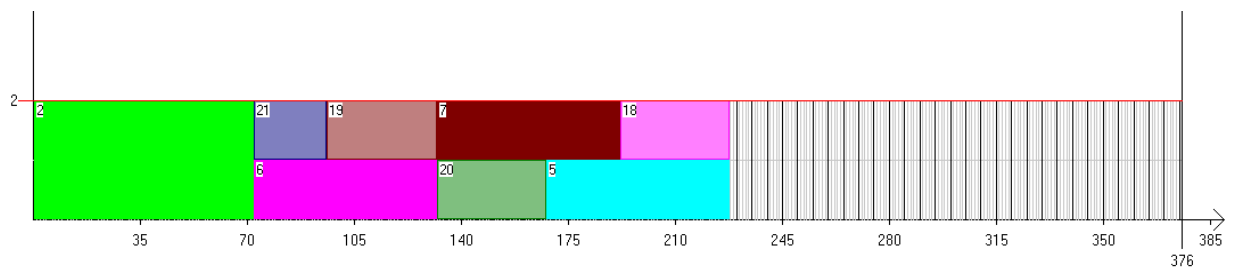
Al tomar los datos del proyecto e ingresarlos al sistema RESCON obtenemos el siguiente diagrama, que representa las actividades, su relación con otras actividades y los recursos asociados a cada una.



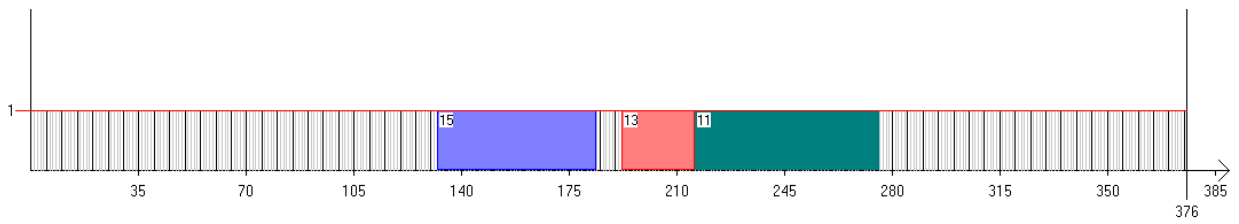
Seguidamente, se presenta la óptima asignación de recursos tipo 2, tipo 3 y tipo 4, según la lógica del algoritmo *Branch & Bound*, así como la Carta Gantt del proyecto optimizado.



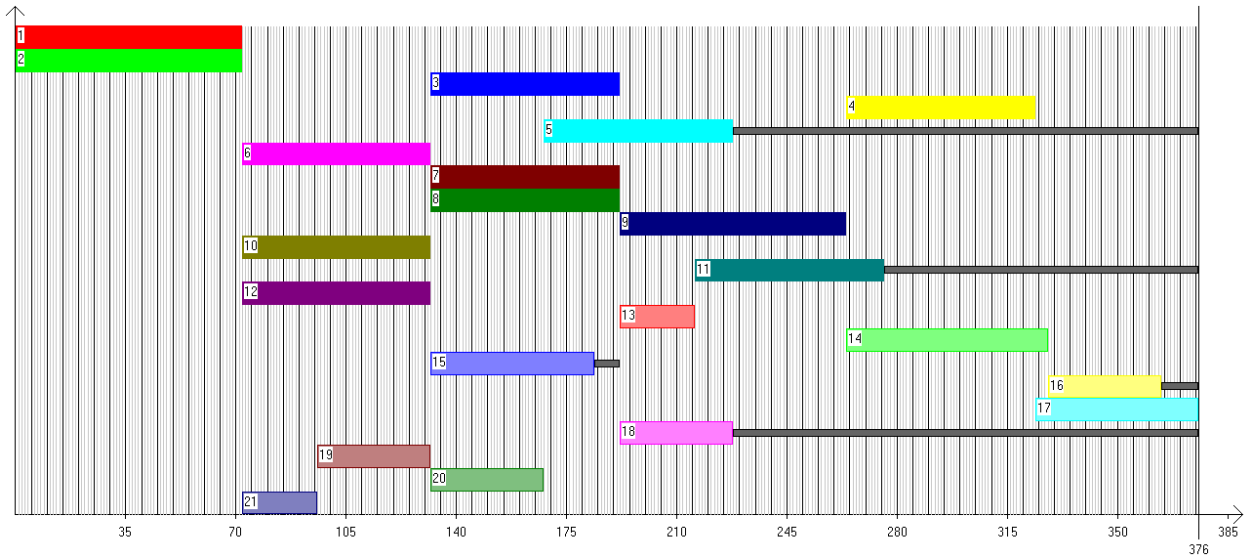
Asignación de recursos tipo 2



Asignación de recursos tipo 3



Asignación de recursos tipo 4



Carta Gantt

Los resultados obtenidos determinan que el proyecto pudo haberse ejecutado en un plazo máximo de 376 horas-hombre, equivalente a 47 días hábiles aproximadamente. Este resultado genera un ahorro de 10 días hábiles, equivalente a un 18% del tiempo empleado en el proyecto real.

También es posible apreciar que los recursos tipo 2, quedan libres transcurridas 230 horas desde el inicio del proyecto, lo cual permitiría trasladarlos a nuevos proyectos que requieran expertos con las capacidades tipo 2. De forma similar los recursos tipo 4 sólo se necesitan durante 136 horas consecutivas, por lo que el resto del tiempo podrían apoyar otros proyectos.

A continuación se presenta una tabla comparativa entre el proyecto real y el proyecto optimizado.

Días usados Real	HH usados Simulación	Ahorro (días)
57	47	10

	Ing. Requeridos Real	Ing. Requeridos Simulación	Ahorro (personal)
<b>Tipo 2</b>	2	2	0
<b>Tipo 3</b>	2	2	0
<b>Tipo 4</b>	1	1	0

En conclusión se puede afirmar que sí es posible realizar una asignación optimizada que permita una mejor distribución de las actividades. Lo cual conlleva al máximo aprovechamiento de la capacidad de los ingenieros durante el mayor tiempo posible. Así mismo, y según los resultados obtenidos, se ha demostrado que es posible determinar el “tiempo libre” de los ingenieros y poder reasignarlos a otros proyectos, minimizando la necesidad de contratar nuevos ingenieros por cada nuevo proyecto que se ejecute.

## **9. DESARROLLO DEL PROTOTIPO**

Una vez diseñados los procesos, es fundamental brindarle a los usuarios una plataforma que los soporte y les de facilidades para registrar, controlar y darle seguimiento a sus proyectos, que como ya se ha mencionado, antes el diseño de los procesos no era estructurada y carecía de buenas prácticas. Como bien se ha fundamentado en el Marco Teórico, se han seguido las pautas enunciadas por el PMI (*Project Management Institute*), organización que se ha vuelto un estándar en lo que a administración de proyectos se refiere, para el desarrollo de dicho sistema.

En este capítulo se muestran capturas de pantalla del prototipo que se ha desarrollado para el ingreso, control y gestión de proyectos de telecomunicaciones para Controlca S.A. El piloto ha sido concebido para gestionar sólo proyectos de ingeniería básica, los cuales representan la mayoría de los tipos de proyectos ejecutados. Este prototipo hace especial énfasis en la gestión de recursos que pudiesen ser utilizados en otros proyectos. La finalidad del prototipo es el control, planificación y seguimiento de proyectos buscando maximizar el uso de los recursos y minimizar la contratación de personal para nuevos proyectos.



El desarrollo de este prototipo ha fomentado nuevas iniciativas para la gestión de proyectos, las cuales serán implementadas a futuro como una mejora y apoyo al sistema base. El éxito de este prototipo, en cuanto a su utilidad y estrategia de desarrollo modular, determinó que el sistema fuera ampliado para incluir el control y planificación de proyectos de ingeniería de detalle y FEED.

### 9.1.1. Control de Proyectos

En la siguiente figura: "Sistema de Ingreso" se puede apreciar un control de acceso, que sirve para aceptar sólo usuarios registrados al sistema, debido a las restricciones de seguridad y privacidad que deben estar implícitas en cualquier sistema de carácter estratégico, o que contenga información estratégica de una empresa. Este sistema de acceso, también permitirá definir dos perfiles de usuario, tales como: Administrador y Usuario.

El sistema de ingreso, además cuenta con un sistema anti-robots tipo *Captcha*, el cual ofrece un nivel de seguridad adicional. Este se emplea para determina si el usuario es o no humano. En caso de detectarse un robot, la IP origen es bloqueada permanentemente.

PLATAFORMA DE GESTIÓN DE PROYECTOS - CONTROLCA S.A. [\[ Iniciar Sesión \]](#)

Ingreso de Proyectos Planificación de Proyectos Control de Proyectos

**INICIO DE SESIÓN**  
Por favor introduzca su nombre de usuario y contraseña.

Información de la cuenta

Usuario:  
ateixeira

Contraseña:

Captcha:

**Iniciar Sesión**

### 9.1.2. Listado y Creación de Proyectos

Se ha desarrollado también un panel navegable en donde se despliega el listado de proyectos. Además del código del proyecto y su nombre, se muestra el estado global en el que se encuentra. Los estados pueden ser: Atrasado, En ejecución, Pendiente por Aprobación, Rechazado, Suspendido y Finalizado. El panel con algunos proyectos de muestra se pueden visualizar en la siguiente figura: "Listado de Proyectos".



The screenshot shows a web application interface for project management. At the top, it says 'PLATAFORMA DE GESTIÓN DE PROYECTOS - CONTROLCA S.A.' and 'Bienvenido, Alejandro Teixeira [Cerrar Sesión]'. Below this are navigation tabs: 'Nuevo Proyecto', 'Listado de Proyectos', and 'Planificación'. Under 'Listado de Proyectos', there are buttons for 'Resumen' and 'Generar CDP'. The main content is a table with the following data:

idCDP	tCDP	usuario	estado	comentario	Nombre
16022011160651	Basica	Alejandro Teixeira	Suspendido	En espera de información	Petrozulia
16022011165010	Basica	Luis Serpa	Rechazado	CDP mal generado	Petrozulia
16022011173134	Basica	Alejandro Teixeira	En ejecución		PDVSA
17022011142212	Basica	Alejandro Teixeira	Atrasado	En espera de TEC	Laraisa
17022011144758	Basica	Alejandro Teixeira	En ejecución		Conversión Profunda fase I
17022011150952	Basica	Luis Serpa	Pendiente por Aprobación		Conversión Profunda fase II
17022011151033	Basica	Luis Serpa	En ejecución		El Furrial Crudo

Para crear un proyecto nuevo en la plataforma se debe ingresar a una interfaz como la que se muestra en la siguiente figura: "Crear Ficha Proyecto", en la cual el administrador deberá indicar los parámetros característicos del proyecto, los cuales definirán los productos asociados al proyecto y la carga horaria que cada uno tendrá. Tanto la definición de los productos como su estimación en horas se calculan en base a reglas de negocio pre-establecidas en el sistema. Esas reglas de negocio se han levantado empíricamente y materializadas en el sistema.

PLATAFORMA DE GESTIÓN DE PROYECTOS - CONTROLCA S.A. [Bienvenido, Alejandro Teixeira [Cerrar Sesión](#)]

Nuevo Proyecto | Listado de Proyectos | Planificación

Resumen | Generar CDP

Tipo de Proyecto:

Idioma:

Fecha de Inicio: 

April 2012						
Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	31
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	1	2	3	4	5

Duración (semanas):

Factor de Riesgo:

Activar	id	Categoría
<input checked="" type="checkbox"/>	A	Cableado en Cobre y Fibra
<input type="checkbox"/>	B	Radiocomunicaciones
<input checked="" type="checkbox"/>	C	Networking
<input type="checkbox"/>	D	Comunicaciones
<input type="checkbox"/>	E	Control y Seguridad

Localización:

**Sistemas**

- CABLEADO ESTRUCTURADO
- CCTV
- CONDUITS Y CANALIZACIONES INTERNAS
- CONTROL DE ACCESO

Para la creación de un nuevo proyecto el administrador deberá definir los siguientes parámetros: Categorías y Sub-Categorías de Sistemas de Telecomunicaciones, Idioma del proyecto, Localización del Proyecto, Factor de riesgo, Tipo de Proyecto (para el piloto solo se considero: Ingeniería Básica), Complejidad de Edificios y Zonas, Inicio Esperado, Final Acordado.

Una vez generado el proyecto, el sistema desplegará el listado de productos de ingeniería que deberán ejecutarse durante el proyecto, junto al valor estimado de horas que deberán ser consumidas para su elaboración. A continuación se presenta una figura que ejemplifica el CDP (control de proyectos) de un proyecto.

PLATAFORMA DE GESTIÓN DE PROYECTOS - CONTROLCA S.A. [Bienvenido, Alejandro Teixeira [Cerrar Sesión](#)]

Nuevo Proyecto   Listado de Proyectos   Planificación

Resumen   Generar CDP

**CONTROL DE PROYECTOS**  
Sistema de generación automática de productos

Total de HH: **48,788**   [Descargar](#)   [Guardar](#)

**ACTIVIDADES DE ING.BASICA**

CIN	tAct	Actividad	P2	P4	P8	P9
W04		APOYO INTERDISCIPLINARIO	0	456	0	56
W00		COORDINACION	0	0	0	512
W06		INSPECCION DE CAMPO	192	192	0	8
W06		LEVANTAMIENTO DE INFORMACION	256	256	0	0
W01	ING	LIBROS DE INGENIERIA	320	320	0	48
W09		MESAS DE TRABAJO CON EL CLIENTE	0	0	0	72
W01	ING	REVISIÓN DE LA INGENIERÍA CONCEPTUAL	0	320	0	80
W09		REVISION TECNICA	0	0	35148	0
		SOPORTE A ELABORACIÓN DOCUMENTO DSO	0	0	0	80

Una vez generado el CDP el proyecto queda en estado "Pendiente por Aprobación", esperando a que Inelectra (cliente directo) lo apruebe.

### 9.1.3. Planificación de Proyectos

El módulo de Planificación de Proyectos es quizás el más importante, dado que es aquí donde se determina el grupo de trabajo ideal para la ejecución del proyecto. Para ello el sistema muestra en pantalla los perfiles de experticia que se requieren para la ejecución del proyecto y despliega para cada caso aquellos ingenieros que posean disponibilidad horaria superior al 10%. El administrador deberá seleccionar sus prioridades a modo de que el sistema los tome en cuenta al momento de realizar las asignaciones de tareas. El panel de planificación se muestra en la siguiente figura.

PLATAFORMA DE GESTIÓN DE PROYECTOS - CONTROLCA S.A. [ Iniciar Sesión ]

Nuevo Proyecto   Listado de Proyectos   Planificación

ID Proyecto: 17022011144758  
 Nombre del Proyecto: Conversión Profunda fase I  
 Jefe de Proyecto: Eduardo Stiasni  
 Total HH: 9.255

**Categorías**

Activar	id	Categoría
<input checked="" type="checkbox"/>	A	Cableado en Cobre y Fibra
<input type="checkbox"/>	B	Radiocomunicaciones
<input checked="" type="checkbox"/>	C	Networking
<input checked="" type="checkbox"/>	D	Comunicaciones
<input checked="" type="checkbox"/>	E	Control y Seguridad

**Ingenieros**

Activar	Nombre	Experticia	Disponibilidad
<input checked="" type="checkbox"/>	Andres Caraballo	A,D,E	33%
<input checked="" type="checkbox"/>	Gustavo Angulo	A,B,C,D	17%
<input checked="" type="checkbox"/>	Alexander Henriques	A,C,D	59%
<input checked="" type="checkbox"/>	Luis Eduardo Serpa	A,B,C,D,E	25%
<input type="checkbox"/>	Aurora Armesto	A,B,C,D,E	36%
<input checked="" type="checkbox"/>	Patricia Graffe	A	85%
<input type="checkbox"/>	Ana Ligia Parra	A,E	77%

Este módulo busca maximizar el uso de horas hombre disponible en recursos calificados con el fin de minimizar la contratación de nuevo personal, es por ello que al momento de autorizar la ejecución de la planificación el sistema devuelve como resultado una tabla con indicadores que comparan la solución óptima, calculada en base a la disponibilidad, perfil del ingeniero y tipos de tareas, con la solución "preferente", la cual toma en cuenta los usuarios pre-seleccionados por el administrador. A continuación se presenta una figura que detalla los resultados obtenidos en cada caso.

PLATAFORMA DE GESTIÓN DE PROYECTOS - CONTROLCA S.A. [ Iniciar Sesión ]

Nuevo Proyecto    Listado de Proyectos    Planificación

ID Proyecto: 17022011144758  
Nombre del Proyecto: Conversión Profunda fase I  
Jefe de Proyecto: Eduardo Stiasni  
Total HH: 9.255  
Duración del Proyecto: 24 semanas  
HH autorizadas por semana: 400

**Asignación Preferencial**

Categoría	HH Requeridas	ING	HH Útiles	HH Restantes	Eficiencia
A	3.240	PG	720	2.520	2,63
C	2.314	AH	470	1.844	1,92
D	926	GA	67	859	0,99
E	2.775	AC/LS	364	2.411	2,51

Eficiencia general: 1,15

Contratar:  
- Categoría A: 3  
- Categoría C: 2  
- Categoría D: 1  
- Categoría E: 3

**Asignación Optimizada**

Categoría	HH Requeridas	ING	HH Útiles	HH Restantes	Eficiencia
A	3.240	JR/PG	1.468	1.771	1,85
C	2.314	AH	470	1.844	1,92
D	926	SR/ED/LS	1.046	0	-0,13
E	2.775	AA/AP	892	1.882	1,96

Eficiencia general: 1,07

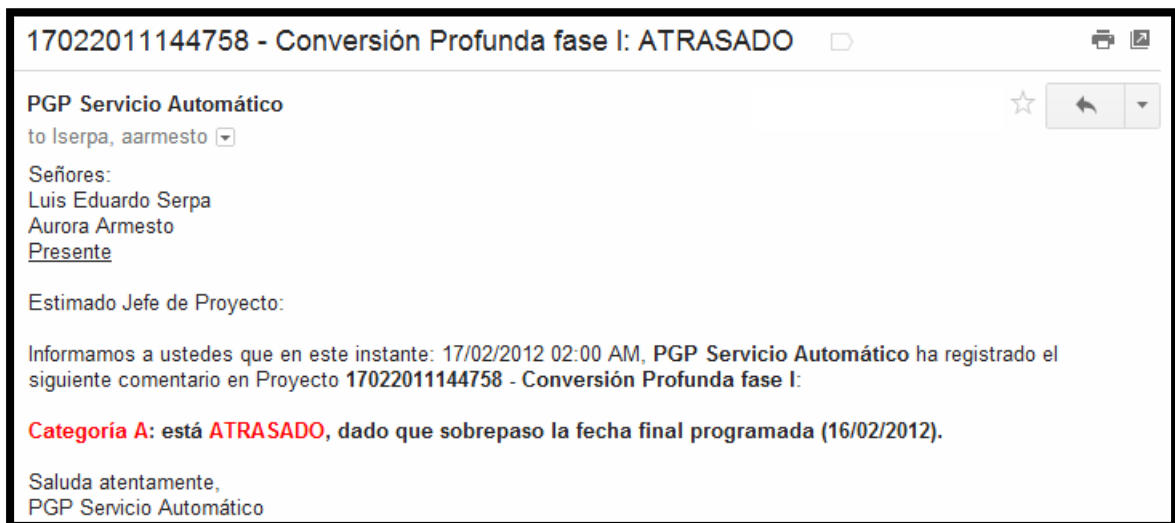
Contratar:  
- Categoría A: 2  
- Categoría C: 2  
- Categoría E: 2

En caso de que el total de horas-hombre supere el total disponible, entonces el sistema informará la cantidad de personas sugeridas para contratación, así como el perfil mínimo de conocimiento que debería poseer (categoría).

El administrador debe seleccionar su grupo de trabajo y en caso de ser necesario, solicitar a Recursos Humanos la contratación de nuevos ingenieros. Estos ingenieros deben cumplir como mínimo con el nivel del experticia requerido para el proyecto (categorías).

#### 9.1.4. Sistema de Alertas

Los proyectos, tareas y actividades manejan perfectamente rangos de vencimiento y generan alertas tempranas ante vencimientos. El sistema emite estas alertas a través de correos electrónicos a los usuarios y notificaciones en el sistema para el administrador, quien se encarga de colaborar con los usuarios responsables en las actividades que sean necesarias para llevar a cabo los compromisos. En la siguiente figura: "Alerta" se puede apreciar una captura de pantalla de un correo que notifica una alerta por atraso.



### 9.1.5. Actualización de Proyectos

Una vez que un proyecto es almacenado, puede ser consultado posteriormente, ofreciendo la posibilidad de actualizar y modificar los plazos de entrega, así como la incorporación o desincorporación de actividades al CDP. Es evidente que esto tiene un efecto directo sobre la disponibilidad horaria que puedan tener los diferentes integrantes del proyecto. Es por esto que cualquier cambio implica el re-cálculo de las horas hombre del proyecto. Este hecho podría afectar la incorporación o desincorporación de personal en el proyecto.

### 9.1.6. Autogeneración de cartas Gantt

La plataforma en base a la información suministrada por el administrador y en base a reglas de negocio prepara una carga Gantt del proyecto. Esta planificación agrupa los productos en categorías, mostrando la duración del conjunto durante el tiempo de vida del proyecto. A continuación se muestra un ejemplo de carta Gantt de un proyecto.



### 9.1.7. Panel de Control

El sistema cuenta con un Panel de Control que permite visualizar reportes, reportar uso de horas hombre y una bitácora por proyecto que deja constancia de los cambios que se han efectuado. Además permite notificar al grupo del proyecto sobre avances o retrasos del proyecto.

Los reportes arrojados por el sistema son los siguientes:

- Reporte de Avance Semanal.
- Estado de los Proyectos.
- Estado de Ocupación
- Indicadores de Gestión por Proyecto

A continuación se presenta un reporte que indica el estado de ocupación de los ingenieros de proyecto.



PLATAFORMA DE GESTIÓN DE PROYECTOS - CONTROLCA S.A. [ Iniciar Sesión ]

Nuevo Proyecto   Listado de Proyectos   Planificación

ESTADO DE OCUPACIÓN

**Ingenieros de Proyecto Fijos**

ING	Categorías	HH Libres (%)	Fecha Ocupación
AA	ABCDE	18%	04/2012
AP	AE	67%	02/2013
SR	ABCDE	67%	08/2013
MZ	AE	11%	03/2012
JR	AC	70%	12/2013
ED	ABCDE	43%	12/2012
AC	ADE	19%	10/2013
GA	ABCD	15%	04/2013
AH	ACD	57%	12/2012
LS	ABCDE	13%	08/2012
PG	A	73%	08/2013

**Lideres de Proyecto:**  
 Lideres de Proyecto:  
 - ED: Eduardo Stiassni  
 - LS: Luis Serpa  
 - AA: Aurora Armesto

**Proyectos Activos**  
 - PDVSA  
 - Conversión Profunda fase I  
 - El Furrial Crudo

El sistema cuenta con una bitácora, la cual sirve como un libro de comentarios por proyecto que registra información de los usuarios y la sube al sistema. Estos comentarios quedan disponibles para que cualquier usuario los revise y se mantenga al tanto de los avances del proyecto. Cada vez que un usuario agrega un comentario, una notificación es enviada por correo electrónico a los integrantes del proyecto. De alguna manera sirve como una especie de sistema de gestión documental. A continuación se presenta una figura que ejemplifica el Libro de Comentarios de un Proyecto.

PLATAFORMA DE GESTIÓN DE PROYECTOS - CONTROLCA S.A. [ Bienvenido, Alejandro Teixeira Cerrar Sesión ]

Nuevo Proyecto   Listado de Proyectos   Planificación

Resumen   Generar CDP

BITACORA DE PROYECTOS

Mensaje:

Emoticones:

**Eduardo Stiassni**

Estimados,

Se notifica que ha habido un cambio de alcance para los Sistemas de Control de Acceso en Conversión Profunda.

Saludos.

Fecha: 02/02/2012 | 05:27 PM

## 10. GENERALIZACIÓN DE LA EXPERIENCIA

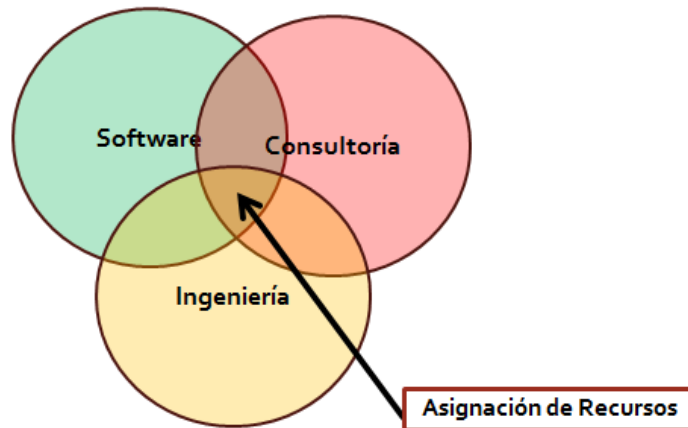
En esta sección, se pretende explicar el patrón de diseño, a nivel de proceso y computacionalmente, a modo de que el mismo pueda ser utilizado en otros proyectos.

Dado el contexto en el que se desarrolla este proyecto:

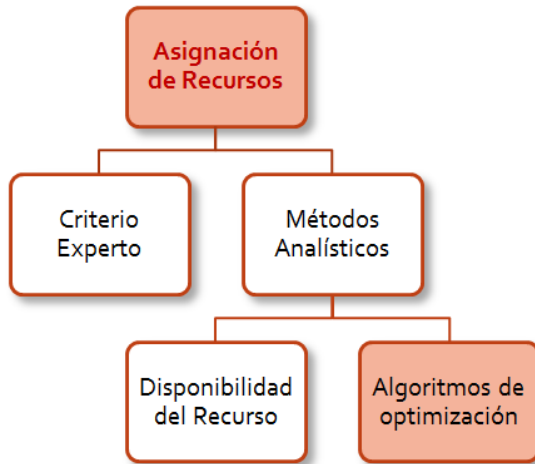
- Planificación de proyectos de Telecomunicaciones
- Gestión de proyectos de Telecomunicaciones

Es posible definir el siguiente dominio:

- Consultoría
- Desarrollo de Software
- Ingeniería



A continuación se presenta la estructura del problema de forma genérica.



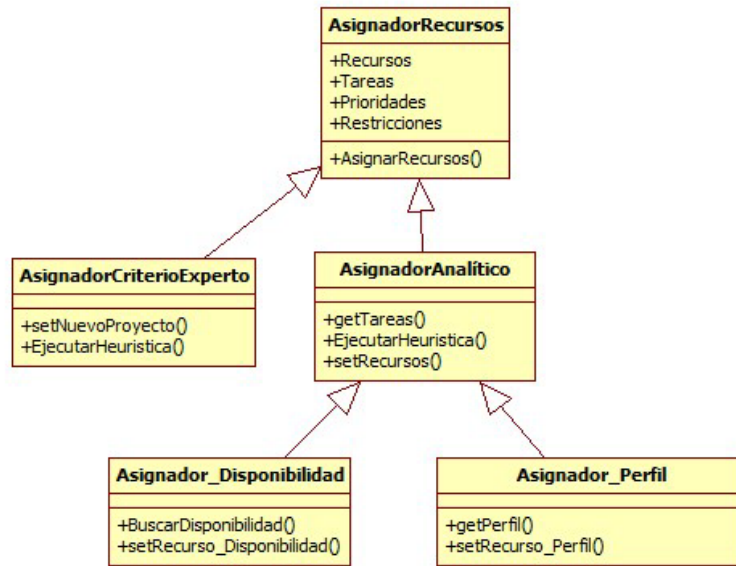
Los **objetos del sistema** que se pueden personalizar para adaptarse a una situación específica son los siguientes:

- Métodos analíticos
- Algoritmos de optimización

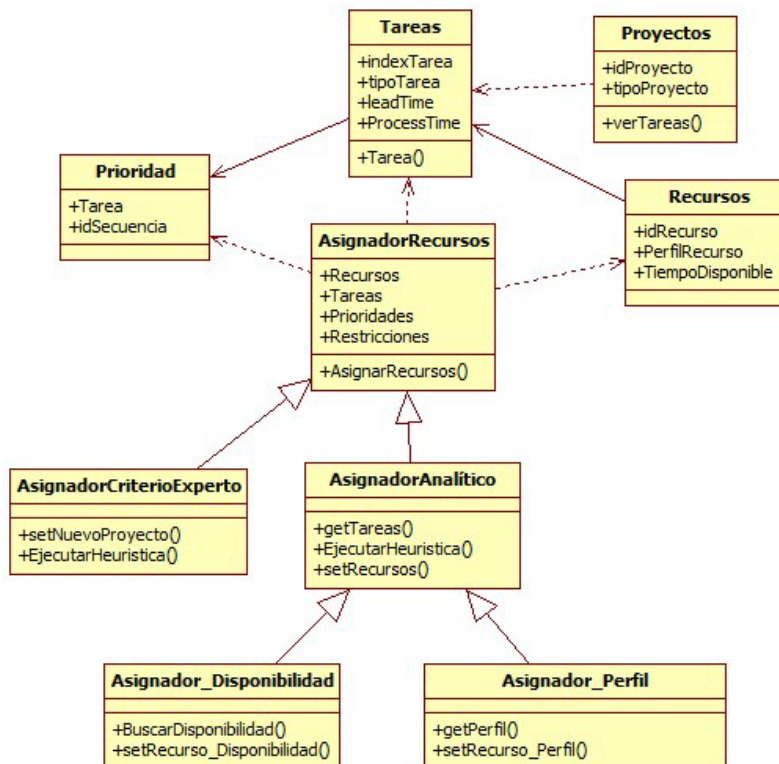
Estos objetos varían según el dominio al cual se aplica el sistema. Por lo tanto, el sistema deberá estar construido de forma tal que permita el remplazo de dichas clases por paquetes de clases de especialización de dominio.

Las clases reemplazables se deben definir mediante una jerarquización abstracta que le permita al sistema cargar y ejecutar la clase específica, sin tener que modificar el código fuente.

A continuación se presentan los *Business Object* referentes al dominio descrito.



Seguidamente, se presenta un framework, basado en los *Business Object* que representa el diagrama de clases de la asignación de recursos, incluyendo la relación de los objetos con los *entity* y métodos.



Algunas consideraciones técnicas que se deben tener en cuenta para lograr una óptima especialización del sistema:

- Parametrización de las clases de negocio. De esta forma se implementa un centro de control para la configuración de las clases de negocio a cargar.
- Métodos genéricos y fáciles de intervenir

Los beneficios obtenidos mediante este diseño son de suma importancia, ya que se logra personalizar el dominio del negocio de la empresa en cuestión, sin intervenir el sistema. Algo mucho más potente es la situación de separar la lógica específica de un negocio, del resto de la lógica del sistema. Esto último permite que el desarrollo de la primera sea realizada por el departamento de informática de la empresa en donde se implementará el sistema, consiguiendo guardar exclusividad y reserva de estas clases de negocio, al no compartirlas con el proveedor del sistema, protegiendo así su know-how.

## **11. PERSPECTIVAS FUTURAS DEL PROYECTO DE GRADO**

La idea inicial del proyecto era la de un ejercicio intelectual que demostrara que, empleando metodologías de optimización, era posible mejorar sustancialmente la asignación de tareas y el uso del personal en los proyectos de telecomunicaciones llevados a cabo por la empresa Controlca.

A medida que se fue desarrollando este trabajo, los alcances fueron cambiando y también la visión a futuro de la empresa en cuanto al uso práctico de dicho proyecto. Dados los resultados obtenidos, y al ser contrastados con los datos reales, surgió la necesidad de mejorar la metodología implementada por la empresa hasta el momento. Este proceso se desarrolló basándose en los estudios y levantamiento de procesos realizados en este proyecto de grado, para dar inicio a lo que sería el replanteamiento en la asignación de tareas al personal calificado de la empresa.

Es por ello, que este proyecto no solo queda como ejercicio intelectual, ni como una prueba de concepto, sino que será tomado como base para el desarrollo de metodologías que permitan optimizar el uso de las horas hombre, así como la asignación de recursos y la planificación de los proyectos.

## 12. **BIBLIOGRAFIA**

1. Algoritmo heurístico híbrido con múltiples vecindarios y recocido simulado para resolver el RCPSP, Consultada en Julio de 2012  
[http://www.scielo.unal.edu.co/scielo.php?script=sci\\_arttext&pid=S0120-62302010000600025&lng=es&nrm=&tlng=es](http://www.scielo.unal.edu.co/scielo.php?script=sci_arttext&pid=S0120-62302010000600025&lng=es&nrm=&tlng=es)
2. Estrategias heurísticas adaptativas para la solución del problema de programación de proyectos con recursos limitados (RCPSP) , Consultada en Julio de 2012  
[http://www2.unalmed.edu.co/minas/index2.php?option=com\\_docman&task=doc\\_view&gid=885&Itemid=57](http://www2.unalmed.edu.co/minas/index2.php?option=com_docman&task=doc_view&gid=885&Itemid=57)
3. Problem-centric scheduling for heterogeneous computing systems, Consultada en Agosto de 2012  
[http://sydney.edu.au/engineering/it/research/2008\\_Young\\_Lee\\_thesis.pdf](http://sydney.edu.au/engineering/it/research/2008_Young_Lee_thesis.pdf)
4. Nuevos Métodos Meta Heurísticos para la Asignación Eficiente, Optimizada y Robusta de Recursos Limitados, Consultada en Agosto de 2012  
[http://sydney.edu.au/engineering/it/research/2008\\_Young\\_Lee\\_thesis.pdf](http://sydney.edu.au/engineering/it/research/2008_Young_Lee_thesis.pdf)
5. An Approach to Resource Constrained Project Scheduling, Consultada en Agosto de 2012  
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.53.5222>
6. GPU implementations of scheduling heuristics for heterogeneous computing environments, Consultada en Julio de 2012  
[http://www.fing.edu.uy/inco/cursos/hpc/material/clases/gpu\\_hcsp\\_heuristics-presentation.pdf](http://www.fing.edu.uy/inco/cursos/hpc/material/clases/gpu_hcsp_heuristics-presentation.pdf)
7. Resource-constrained project scheduling: Notation, classification, models, and methods  
[http://ie406.cankaya.edu.tr/uploads/files/EJOR\\_free26.pdf](http://ie406.cankaya.edu.tr/uploads/files/EJOR_free26.pdf)
8. Making project management indispensable for business results, 2012, Consultada en Enero de 2012, [www.pmi.org](http://www.pmi.org)
9. Introduction to PMBOK Guide and standards, 2012, Consultada en Enero de 2012, [www.pmi.org](http://www.pmi.org)

10. PMBOK Guide and the library of PMI standards, 2012, Consultada en Enero de 2012, [www.pmi.org](http://www.pmi.org)
11. Estructura del PMBOK, Consultada en Diciembre de 2011, [http://liderdeproyecto.com/manual/estructura\\_del\\_pmbok.html](http://liderdeproyecto.com/manual/estructura_del_pmbok.html)
12. La Ingeniería de Negocios, 2009, Consultada en Julio 2011, [http://www.docirs.cl/ingenieria\\_negocios\\_hernandezdemedina.htm](http://www.docirs.cl/ingenieria_negocios_hernandezdemedina.htm)
13. Análisis Porter de las cinco fuerzas, 2010, Consultada en Julio 2011, [http://es.wikipedia.org/wiki/An%C3%A1lisis\\_Porter\\_de\\_las\\_cinco\\_fuerzas](http://es.wikipedia.org/wiki/An%C3%A1lisis_Porter_de_las_cinco_fuerzas)
14. El Modelo Delta un Nuevo Marco Estratégico, 2003, Consultada en Junio 2011, Arnoldo C. Hax, MIT
15. Arnoldo Hax sobre la relación del Modelo Delta con los "strategic frameworks" más tradicionales, 2008, Consultada en Junio 2011
16. <http://innovacionygestion.bligoo.com/content/view/208497/Arnoldo-Hax-sobre-la-relacion-del-Modelo-Delta-con-los-strategic-frameworks-mas-tradicionales.html>
17. Ing. de Negocios con TI, 2011, Consultada en Abril 2011, <http://blog.obarros.cl/archives/category/ingenieria-de-negocios-con-ti>



# **ANEXO A**

# An Approach to Resource Constrained Project Scheduling

James M. Crawford

CIRL

1269 University of Oregon

Eugene, OR 97403-1269

jc@cirl.uoregon.edu

## Abstract

This paper gives an overview of a new approach to Resource Constrained Project Scheduling. The approach is based on the combination of a novel optimization technique with limited discrepancy search, and generates the best known solutions to benchmark problems of realistic size and character.

## Introduction

Historically there has often been a mismatch between the types of scheduling problems that have been studied academically and the needs of the manufacturing community. The most obvious difference is that many real problems are much larger than common academic benchmarks. A second, and equally important, difference is that real problems generally involve constraints that have a more complex structure than can be expressed within a limited framework like job shop scheduling.

In this paper we overview ongoing work on resource constrained project scheduling (RCPS). RCPS is a generalization of job shop scheduling in which tasks can use multiple resources, and resources can have a capacity greater than one. RCPS is thus a good model for problems, like aircraft assembly, that cannot be expressed as job shop problems. In fact, if we take arguably the most widely used commercial scheduling program, Microsoft Project, RCPS seems to capture exactly the optimization problem that the “resource leveler” in Project solves.

It turns out that the algorithms that have been developed for job shop scheduling do not work particularly well for RCPS. In this paper we overview an approach to RCPS that is based on the combination of limited discrepancy search (LDS) with a novel optimization technique. The resulting system produces the best known schedules for problems of realistic size and character.

## Resource Constrained Project Scheduling

A Resource Constrained Project Scheduling (RCPS) problem consists of a set of tasks, and a set of finite capacity resources. Each task puts some demand on the resources. For example, changing the oil might require one workman and one car lift. A partial ordering on the tasks is also given specifying that some tasks must precede others (*e.g.*, you have to sand the board before you can paint it). Generally the goal is to minimize makespan without violating the precedence constraints or over-utilizing the resources.

RCPS is more general than job shop because resources can have capacity greater than one, and because tasks can use a collection of resources. This allows resources to be taken to be anything from scarce tools, to specialized workmen, to work zones (such as the cockpit of an airplane). RCPS problems arise in applications ranging from aircraft assembly to chemical refining.

Formally, Resource Constrained Project Scheduling is the following:

**Given:** A set of tasks  $T$ , a set of resources  $R$ , a capacity function  $C : R \rightarrow \mathbb{N}$ , a duration function  $D : T \rightarrow \mathbb{N}$ , a utilization function  $U : T \times R \rightarrow \mathbb{N}$ , a partial order  $P$  on  $T$ , and a deadline  $d$ .

**Find:** An assignment of start times  $S : T \rightarrow \mathbb{N}$ , satisfying the following:

1. Precedence constraints: if  $t_1$  precedes  $t_2$  in the partial order  $P$ , then  $S(t_1) + D(t_1) \leq S(t_2)$ .
2. Resource constraints: For any time  $x$ , let  $running(x) = \{t | S(t) \leq x < S(t) + D(t)\}$  Then for all times  $x$ , and all  $r \in R$ ,  $\sum_{t \in running(x)} U(t, r) \leq C(r)$ .
3. Deadline: For all tasks  $t : S(t) \geq 0$  and  $S(t) + D(t) < d$ .

## A Benchmark Problem

Our experiments have been run on a series of problems made available on the WWW at:

<http://www.neosoft.com/~benchmrx>

by Barry Fox of McDonnell Douglas and Mark Ringer of Honeywell, serving as Benchmarks Secretary in the AAAI SIGMAN and in the AIAA AITC, respectively. These problems have 575 tasks and 17 resources. Some of the resources represent zone (geometric) constraints, and some represent labor constraints. Labor availability varies by shift. This is a synthetic problem that has been generated from experience with multiple large scale assembly problems. It is comparable to real problems in size and character, but simpler in the complexity of the constraints.

The results that have been posted to the WWW to date are shown in figure 1. Mark Ringer's results were found using a simple, first fit, interval based algorithm with no optimization. They were posted to encourage other contributions, rather than to generate the best solutions possible.

Who	Problem			Note
	2	3	4	
Mark Ringer	45	57	56	Honeywell
Nitin Agarwal	40	47	57	SAS
Colin Bell	39	-	-	Univ. of Iowa
Barry Fox	38	45	42	McDonnell Douglas
Crawford <i>et. al.</i>	38	43	41	CIRL
Crawford <i>et. al.</i>	38	39	38	(Lower Bound)

Figure 1: Results

## Solution Methods

The two most important methods used in our scheduler are doubleback optimization and limited-discrepancy search. We discuss each in turn and then discuss how they work together in the scheduler.

### Doubleback Optimization

The optimizer starts with any schedule satisfying the precedence constraints and generates a legal (and often a shorter) schedule. It works in two steps: a right shift and then a left shift (a very similar technique, *schedule packing* was independently invented previously by Barry Fox [1996]).

We first establish the right hand end point. Recall that the availability of some labor resources varies by shift. Because of this it turns out that it matters where in the daily cycle the endpoint is set.<sup>1</sup> The best heuristic seems to be to set it at the same point in the daily

<sup>1</sup>This was first observed by Joe Pemberton.

cycle as the end point of the current schedule. Another approach is to select the right hand end point randomly. This tends to shake things up a bit and makes iterating the optimizer more effective.

Once the right hand endpoint is selected we right shift the schedule. In the right shift we take the tasks in order of decreasing finish times (i.e., from the right hand end). We shift each task as far right as it will go. In doing this shift we consider only precedence constraints and resource constraints with previously shifted tasks (one way to think about this is to envision the new right hand endpoint as being at positive infinity: the tasks that have not yet been shifted do not interfere with the construction of the new schedule).

Once the right shift is completed, we left shift back to time zero, starting from the beginning of the right shifted schedule. As before, we left shift as far as possible subject to precedence constraints, and resource conflicts with previously left shifted tasks.

This sequence can be iterated. At some points this produces longer schedules (possibly then followed by shorter schedules after additional iterations). At present we have no theoretical method to predict the optimal number of iterations: we simply iterate ten times and keep the best schedule produced.

To see the optimization works, consider the example shown in figure 2.

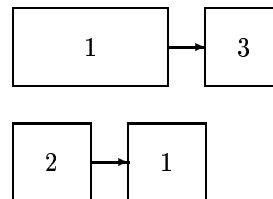


Figure 2: A simple scheduling problem.

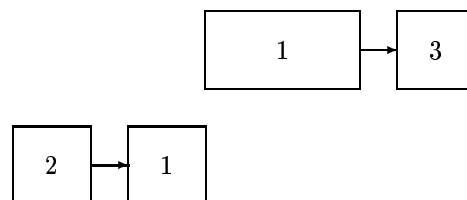


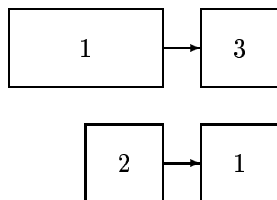
Figure 3: A bad schedule.

Here the boxes represent tasks and the arrows represent precedence relations. The numbers in the boxes are the resources the tasks need. For this example all resources have capacity one.

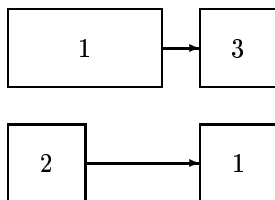
In order to break the resource conflict between the two tasks using resource one, we have to establish an

---

**Right shift:**



**Left shift:**



---

Figure 4: The effect of a right and then left shift.

---

ordering between the tasks. Assume that we do this non-optimally, generating the schedule shown in figure 3.<sup>2</sup>

Now consider what happens when we apply the optimizer. The right-shifted, and then left-shifted, schedules are shown in figure 4. The key thing to notice is that in the right shift the bottom task falls to the end of the schedule (because it has no successors), while the top task is forced to the beginning of the schedule. Thus the left shift schedules the top task first, generating the optimal schedule.

We can, of course, generate test cases in which the optimizer fails to find the shortest schedule, and we currently cannot offer any theoretical guarantees on the optimizer's performance. The strongest statement we can make is that on the benchmark examples, the optimizer is experimentally the single most effective scheduling technique we are aware of.

### Why the Optimizer Works

Experimentally doubleback optimization is quite effective. Starting with the schedule that starts each task as early as possible subject to only the precedence constraints, the optimizer is able to produce 43 day schedules for problem 4. The obvious question is why such a simple technique works so well.

In a sense the key decision to be made in scheduling is the ordering of tasks that compete for a resource. In fact the search portion of our approach (see below) essentially searches all possible ways to break resource

---

<sup>2</sup>Such a mistake is unlikely in such a small problem, but our ability to avoid analogous mistakes in larger problems is, in a sense, the entire source of the intractability of scheduling.

contention by establishing an ordering between competing tasks.

The challenge of breaking resource contention is that we do not know which task will turn out to be the most important. In some cases it is obvious which tasks are most critical. For example, if a task must be followed by a long series of tasks, then clearly we want to give it a high priority – otherwise it will be postponed and the "tail" of tasks that follow it is likely to exceed the deadline. Unfortunately it is not generally this simple (or else scheduling would be tractable). In essence what goes wrong is that we do not know how hard it will be to schedule the sets of tasks that must follow the conflicting tasks. However, if we start from a "seed" schedule, then we can decide with reference to the seed, how hard the subsequent tasks will be, and use this information to make better decisions about task priorities.

It turns out that this is exactly what the optimizer does. The right shift pushes all tasks as late as possible. So, if a task is near the beginning of the right shifted schedule, it is there because it must be followed by a large number of tasks. So it should be given high priority. This is exactly what the left shift does: the left shifted schedule is formed by first schedule the tasks that are near the beginning of the right shifted schedule.

### Limited Discrepancy Search

The results returned by the optimizer are sensitive to the "seed" schedule given to the optimizer. One can construct examples of "bad" schedules that the optimizer cannot correct. In a sense the optimizer is walking down to a kind of local minimum, and the quality of the final schedule depends on where the walk starts.

Our implementation uses LDS (Harvey & Ginsberg 1995) to produce a series of seed schedules that are then passed to the optimizer. Here we give a brief overview of LDS. Details can be found in Harvey and Ginsberg [1995].

Imagine that we have a schedule that satisfies the precedence constraints, but not the resource constraints. The natural way to produce a legal schedule is to iteratively pick a resource conflict, delay one or more tasks long enough to break the conflict, and then propagate these delays through the precedence constraints. This is, in fact, how our current implementation works (starting from the left shifted schedule satisfying only the precedence constraints).

Each conflict that is broken creates a choice point, and breaking a series of conflicts produces a search tree. In the case of the benchmark problems this produces a search tree with a branching factor of about three,

and a depth of about 1000.

The traditional approach to searching such a tree is to use a depth-first search. In a depth-first search, we make a series of decisions until we reach a leaf node in the tree (in this case a leaf node is a schedule satisfying both precedence and resource constraints). We then back up to the last choice point and take the other branch, and follow it to a leaf node. We then back up again, this time to the latest branch point that still has unexplored children. Repeating this we eventually search the entire tree, and are thus guaranteed to find the optimal schedule.

Unfortunately, if the search tree is 1000 nodes deep then a depth-first search will examine only a tiny fraction of the entire search tree (backing up perhaps 10 or 20 nodes). Further, it is reasonable to expect that the choices that will be reconsidered are exactly the choice for which the heuristic is most likely to have made the right decision. To see why this is so, notice that near the top of the search tree there are still many resource conflicts, so the heuristics are working from a "schedule" that is far from legal, so the heuristics are having to guess at how the resolution of these other conflicts will interact with the current conflict. Near the bottom of the tree, however, the schedule is in nearly its final form so the heuristics have good information on which to base their decisions.

As a result, traditional depth-first search is relatively little help on scheduling problems. This has lead many practitioners to either use no search (just following the heuristic and returning the first schedule produced) or to use a local search (which can reconsider any decision at any point).

In LDS we fix a bound on the number of times we will diverge from the heuristic. If that bound is zero then we just produce the single schedule given by always following the heuristics. If the bound is one then we produce a set of schedules generated by ignoring the heuristic exactly once.

The difference between LDS and depth-first search is illustrated in figures 5 and 6. In both search trees the branch preferred by the heuristic is always drawn on the left. In figure 5 the leaves are numbered according to the order in which depth-first search will visit them. Notice that if the heuristic makes a mistake high in the tree, for example, at the first choice point, then depth-first search will have to search half of the search tree before correcting the mistake. In the LDS search tree (figure 6), leaf nodes are labeled according to how many times the path from the root to the leaf diverges from the heuristic (*i.e.*, how many right turns are necessary to reach the leaf). LDS searches the leaves by first searching the leaf marked 0, then all the leaves

marked 1, and so on.

If the heuristic is generally correct, but sometimes makes mistakes (as if generally the case with heuristics) then we can reasonably expect to find good quality schedules by searching the nodes for which the number of divergences is low. If the height of the tree is  $h$ , then the complexity of visiting each node with  $d$  divergences is  $h^d$ . In practice we usually set  $d$  to 1 or 2. This produces much better results than a depth-first search examining the same number of nodes. Further, unlike a local search, LDS is systematic: if we continue to raise  $d$  we are guaranteed to eventually find the optimal schedule.

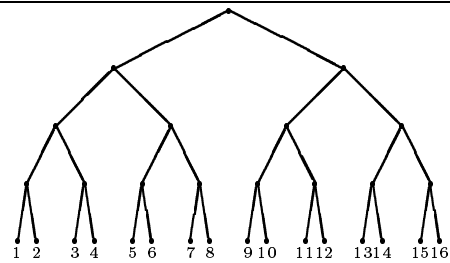


Figure 5: Backtracking search tree.

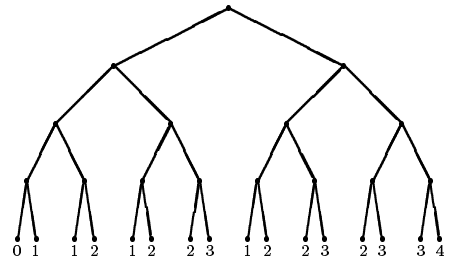


Figure 6: LDS search tree.

Finally we should note that LDS and the optimizer work well together. The optimizer is sensitive to the nature of the schedule it gets as input. Since LDS produces a series of reasonably good (but different) seed schedules, we can optimize each one, and in the end produce a schedule that is significantly shorter than we get by just optimizing the schedule given by following the heuristics exactly.

We can take this one step further and design the heuristic to avoid the kind of mistakes that the optimizer cannot fix.<sup>3</sup> This goes beyond the scope of the current paper, but it turns out that we can identify certain kinds of task-ordering mistakes that a simple right-left shift is unable to untangle. We can then generate heuristics that will generally avoid these mistakes. This may cause us to find worse unoptimized

<sup>3</sup>This idea came out of discussions with Matt Ginsberg.

schedules, but better schedules after optimization.

## Conclusion

We have outlined an approach to RCPS problems that is based on using LDS to generate a series of “seed” schedules that are passed to an optimizer that can be seen as doing a kind of scheduling-specific local search. The results are currently the best known on problems of realistic size and character. Work continues on transitioning this technology to various application areas, and increasing the complexity of the constraints we can represent and effectively optimize under.

## Acknowledgment

This work has been supported by the Air Force Office of Scientific Research under grant number F49620-92-J-0384, by ARPA/Rome Labs under grant numbers F30602-93-C-00031 and F30602-95-1-0023, and by the National Science Foundation under grant number IRI-94 12205. The work was done at the Computational Intelligence Research Laboratory, and owes much to discussions with all the members of the lab, particularly Matt Ginsberg, Joe Pemberton, and Ari Jons-son. Finally, we should add that this work could not have been done without the effort Barry Fox and Mark Ringer have put in to make realistic benchmark problems available on the WWW.

## References

- Fox, B. 1996. An algorithm for scheduling improvement by scheduling shifting. Technical Report 96.5.1, McDonnell Douglas Aerospace - Houston. McDonnell Douglas has applied for a patent on this work.
- Harvey, W. D., and Ginsberg, M. L. 1995. Limited discrepancy search. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, volume 1, 607–613.



ELSEVIER

European Journal of Operational Research 112 (1999) 3–41

EUROPEAN  
JOURNAL  
OF OPERATIONAL  
RESEARCH

Invited Review

## Resource-constrained project scheduling: Notation, classification, models, and methods

Peter Brucker<sup>a,1</sup>, Andreas Drexl<sup>b,\*</sup>, Rolf Möhring<sup>c,2</sup>, Klaus Neumann<sup>d,3</sup>,  
Erwin Pesch<sup>e,4</sup>

<sup>a</sup> *Universität Osnabrück, Fachbereich Mathematik/Informatik, Albrechtstr. 28, 49069 Osnabrück, Germany*

<sup>b</sup> *Universität Kiel, Institut für Betriebswirtschaftslehre, Olshausenstr. 40, 24118 Kiel, Germany*

<sup>c</sup> *Technische Universität Berlin, Fachbereich Mathematik, Straße des 17. Juni, 10623 Berlin, Germany*

<sup>d</sup> *Universität Karlsruhe, Institut für Wirtschaftstheorie und Operations Research, Kaiserstr. 12, 76128 Karlsruhe, Germany*

<sup>e</sup> *Universität Bonn, Institut für Gesellschafts- und Wirtschaftswissenschaften, Adenauerallee 24-42, 53113 Bonn, Germany*

Received 1 June 1998

### Abstract

Project scheduling is concerned with single-item or small batch production where scarce resources have to be allocated to dependent activities over time. Applications can be found in diverse industries such as construction engineering, software development, etc. Also, project scheduling is increasingly important for make-to-order companies where the capacities have been cut down in order to meet lean management concepts. Likewise, project scheduling is very attractive for researchers, because the models in this area are rich and, hence, difficult to solve. For instance, the resource-constrained project scheduling problem contains the job shop scheduling problem as a special case. So far, no classification scheme exists which is compatible with what is commonly accepted in machine scheduling. Also, a variety of symbols are used by project scheduling researchers in order to denote one and the same subject. Hence, there is a gap between machine scheduling on the one hand and project scheduling on the other with respect to both, viz. a common notation and a classification scheme. As a matter of fact, in project scheduling, an ever growing number of papers is going to be published and it becomes more and more difficult for the scientific community to keep track of what is really new and relevant. One purpose of our paper is to close this gap. That is, we provide a classification scheme, i.e. a description of the resource environment, the activity characteristics, and the objective function, respectively, which is compatible with machine scheduling and which allows to classify the most important models dealt with so far. Also, we propose a unifying notation. The second purpose of this paper is to review some of the recent developments. More specifically, we review exact and heuristic algorithms for the single-mode and the multi-mode case, for the time–cost tradeoff problem, for problems with minimum and maximum time lags, for problems with other objectives than makespan minimization and, last but not least, for problems with stochastic activity durations. © 1999 Elsevier Science B.V. All rights reserved.

\* Corresponding author. Fax: +49-431-880-1531; e-mail: [drexl@bwl.uni.kiel.de](mailto:drexl@bwl.uni.kiel.de)

<sup>1</sup> E-mail: [peter@mathematik.uni-osnabrueck.de](mailto:peter@mathematik.uni-osnabrueck.de)

<sup>2</sup> E-mail: [moehring@math.tu-berlin.de](mailto:moehring@math.tu-berlin.de)

<sup>3</sup> E-mail: [neumann@wior.uni-karlsruhe.de](mailto:neumann@wior.uni-karlsruhe.de)

<sup>4</sup> E-mail: [E.Pesch@uni-bonn.de](mailto:E.Pesch@uni-bonn.de)

*Keywords:* Project scheduling/resource constraints; Notation; Classification scheme; Single-mode case; Time–cost tradeoffs; Multi-mode case; Minimum and maximum time lags; Nonregular objectives; Stochastic activity durations; Constraint propagation

---

## 1. Scope and purpose

Project scheduling has attracted an ever growing attention in recent years both from science and practice. It is concerned with single-item or small batch production where scarce resources have to be met when scheduling dependent activities over time. Project scheduling is important for make-to-order companies where the capacities have been cut down in order to cope with lean management concepts. Project scheduling is very attractive for researchers also, because the models in this area are rich in the sense that many well-known optimization problems are special cases of the more general project scheduling models. For instance, the resource-constrained project scheduling problem contains the job shop scheduling problem as a special case. Without surprise, project scheduling problems in general are really challenging from a computational point of view.

Both practice and science of project scheduling have evolved fast recently, producing numerous acronyms to distinguish between different problem classes. Also, a variety of symbols are used by project scheduling researchers in order to denote one and the same subject. Hence, sometimes it is difficult to keep a clear view of what the subject is all about, because the models in this area are not standardized. Recently Herroelen et al. [94] made a first attempt to provide a classification scheme for project scheduling. Unfortunately, their scheme is not compatible with what is commonly accepted in machine scheduling. Hence, there is still a gap between machine scheduling on the one hand and project scheduling on the other with respect to both, viz. a common notation and a classification scheme. One purpose of our paper is to close this gap. We provide a classification scheme, i.e. a description of the resource environment, the activity characteristics, and the objective function, respectively, which is compatible with machine scheduling and which allows to classify the most

important models dealt with so far. Also, we propose a unifying notation.

Another purpose of this paper is to review some of the recent developments.<sup>5</sup> Additional surveys have been given by, e.g., [67,92,113,151].

The paper is organized as follows. In Section 2 the notation and the classification scheme are introduced. Section 3 covers exact and heuristic algorithms for the single-mode resource-constrained project scheduling problem. In Section 4 we review solution procedures for the time–cost tradeoff problem. The multi-mode resource-constrained project scheduling problem is the subject of Section 5. In Section 6 we concentrate on resource-constrained project scheduling in the presence of minimum and maximum time lags. Section 7 is dedicated to problems with nonregular objective functions. Section 8 discusses models with stochastic activity durations. We conclude the paper in Section 9 with an exposition of further models.

Our work will illustrate that there are numerous different models each of which requires tailored methods in order to cope with their inherent (computational) complexity. Nevertheless, constraint propagation techniques have recently evolved, the aim of which is to solve a variety of constraint satisfaction problems. Although their impact on the field of resource-constrained (project) scheduling still is not clear we decided to add an Appendix A the subject of which are constraint propagation-based sequence consistency tests.

## 2. Notation and classification scheme

In the last few years many different problems in project scheduling have been considered and time

---

<sup>5</sup> Most of the working papers covered in this review are available on the internet via <http://www.wior.uni-karlsruhe.de/rcpsp/>.



is ready to have a unified notation and a general classification scheme for project scheduling. It is important that this scheme is compatible with what is generally accepted in machine scheduling (cf. [77]) and resource-constrained machine scheduling (cf. [18]), because machine scheduling models are special cases of project scheduling models.

In the sequel we will first propose a unifying notation. Basically, we assume a project to consist of activities (jobs)  $1, \dots, n$ . For the sake of simplicity, in general a unique dummy beginning activity 0 and a unique dummy termination activity  $n + 1$  are added. Frequently, the structure of the project is depicted by a so-called activity-on-node (AON) network where the nodes and the arcs represent the activities and the precedence relations, respectively.  $G = (V, E)$  denotes the graph of precedence constraints (transitively reduced), while single precedence constraints are denoted alternatively by  $i \rightarrow j$  or  $(i, j)$ .  $Pred(j)$  defines the set of direct predecessors while  $Succ(j)$  is the set of direct successors of activity  $j$ . The processing time of activity  $j$  is given by  $p_j$ .

There is a set  $\mathcal{R}^\rho$  of renewable, a set  $\mathcal{R}^v$  of nonrenewable and, possibly, a set of doubly constrained resources. Renewable means that a pre-specified number of units of a resource is available for every period of the planning horizon  $T$ . Non-renewable says that a number of units of a resource is available for the entire planning horizon. As usual, we skip the notion of doubly constrained resources, because they can be covered by the renewable and the nonrenewable ones. The per period usage of activity  $j$  of renewable resource  $k$  is denoted by  $r_{jk}^\rho$  while  $R_k^\rho$  defines the (constant) number of units of resource  $k$  available in every period. In the multi-mode case,  $\mathcal{M}_j$  defines the set of modes, that is, processing alternatives of activity  $j$ . The processing time of activity  $j$  in mode  $m$  is given by  $p_{jm}$ . The per period usage (total resource consumption) of activity  $j$  of renewable (non-renewable) resource  $k$  is given by  $r_{jkm}^\rho$  ( $r_{jkm}^v$ ) while  $R_k^v$  defines the number of units of nonrenewable resource  $k$  available for the entire planning horizon. In the single-mode case, that is for  $|\mathcal{M}_j| = 1$  for all  $j$  and  $\mathcal{R}^v = \emptyset$ , we skip the mode index  $m$  and the superscript  $\rho$  for the sake of simplicity.

$S_j$  ( $C_j$ ) denotes the start time (completion time) of activity  $j$ . Consequently,  $S = (S_1, \dots, S_n)$  is a schedule and  $C = (C_1, \dots, C_n)$  is the vector of completion times.  $\mathcal{S}_T$  defines the set of time-feasible schedules,  $\mathcal{S}_R$  the set of resource-feasible schedules and  $\mathcal{S} = \mathcal{S}_R \cap \mathcal{S}_T$  the set of feasible schedules.  $t$  is an index for time periods. Finally,  $d_{ij}^{\min}$  and  $d_{ij}^{\max}$  denote minimum and maximum time lags, respectively, between the start of activities  $i$  and  $j$ . In general, parameters are assumed to be integer-valued.

Table 1 summarizes the notation introduced along with some minor additions.

Now we extend the  $\alpha|\beta|\gamma$ -scheme used in the machine scheduling literature.

$\alpha$ : *Resource environment*: To distinguish between specific machine scheduling problems and project scheduling problems we introduce in the  $\alpha$ -field *PS* (project scheduling) or *MPS* (multi-mode project scheduling). *PS* can be augmented to *PS*  $m, \sigma, \rho$  according to the notation of Blażewicz et al. [18] for resource-constrained machine scheduling. In the case of multi-mode project scheduling also nonrenewable resources may be considered. In this case the notation is analogously augmented by *MPS*  $m, \sigma, \rho; \mu, \tau, \omega$ .

<i>PS</i>	project scheduling
<i>MPS</i>	multi-mode project scheduling
<i>PS</i> $m, \sigma, \rho$	$m$ resources, $\sigma$ units of each resource available, each activity requires at most $\rho$ units of the resources
<i>MPS</i> $m, \sigma, \rho; \mu, \tau, \omega$	multi-mode project scheduling with $m$ renewable resources, $\sigma$ units of each resource available, each activity requires at most $\rho$ units of the resources, $\mu$ nonrenewable resources, $\tau$ units of each resource available, each activity requires at most $\omega$ units of the resources

If an entry of  $m, \sigma, \rho; \mu, \tau, \omega$  is replaced by  $\cdot$ , the values of the parameters are specified in the input. For *PS*  $m, \cdot, \cdot$  and *PS*  $m, \sigma, \cdot$  we write *PS*  $m$  and *PS*  $m, \sigma$ , respectively, for short. If all values in  $m, \sigma, \rho$  are specified in the input, we write  $\cdot$  instead of  $\cdot, \cdot, \cdot$ .

Table 1  
Basic notation

Symbol	Definition
$V$	set of activities
$n$	number of real activities
$E$	set of precedence or temporal constraints
$G = (V, E)$	directed graph of precedence or temporal constraints
$i \rightarrow j, (i, j)$	precedence constraint
$Pred(j)$	set of direct predecessors of activity $j$
$Succ(j)$	set of direct successors of activity $j$
$P_j$	processing time of activity $j$
$\mathcal{R}^p$	set of renewable resources
$R_k^p$	constant amount of available units of renewable resource $k$
$r_{jk}^p$	per period usage of activity $j$ of renewable resource $k$
$\mathcal{M}_j$	set of modes (processing alternatives) of activity $j$
$P_{jm}$	processing time of activity $j$ in mode $m$
$\mathcal{R}^n$	set of nonrenewable resources
$R_k^n$	total amount of available units of nonrenewable resource $k$
$r_{jkm}^p$	per period usage of activity $j$ of renewable resource $k$ when processed in mode $m$
$r_{jkm}^n$	consumption of activity $j$ of nonrenewable resource $k$ when processed in mode $m$
$S_j$	start time of activity $j$
$S = (S_1, \dots, S_n)$	schedule
$C_j$	completion time of activity $j$
$C = (C_1, \dots, C_n)$	vector of completion times
$\mathcal{S}_T$	set of time-feasible schedules
$\mathcal{S}_R$	set of resource-feasible schedules
$\mathcal{S} = \mathcal{S}_R \cap \mathcal{S}_T$	set of feasible schedules
$r_k(S, t)$	resource consumption of resource $k$ of schedule $S$ at time $t$
$\bar{d}$	deadline for project duration
$T, t$	time horizon, index for periods
$t = 1, 2, \dots, T$	periods
$[t - 1, t[$	time interval corresponding to period $t$
$d_{ij}^{\min} / d_{ij}^{\max}$	minimum/maximum time lag between start of activities $i$ and $j$

Likewise, for  $PS$  and  $MPS$ ; we write  $PS$  and  $MPS$ , respectively.

*Examples:*

$PSm, 1, 1$	$m$ resources, 1 unit of each resource available, each activity requires at most 1 unit of the resources	$p_j = 1$	all processing times (activity durations) are equal to one
$PSm, \infty$	$m$ resources, unlimited number of resource units available (i.e., there are no explicit resource constraints, e.g. in resource leveling)	$\frac{p_j}{\bar{d}} = sto$	stochastic processing times
$PS1$	one resource	$\bar{d}$	deadline for project duration
		$prec$	precedence constraints between activities
		$chains, intree, outtree, tree \dots$	precedence relations between activities are specified by chains, intree, outtree, tree ...
		$temp$	general temporal constraints given by minimum and maximum start–start time lags between activities

$\beta$ : Activity characteristics: We use established notations from machine scheduling (cf. [77]) like  $p_j$

$\gamma$ : *Objective function*: As in most cases for machine scheduling we describe objective functions by the corresponding formulas. Besides classical objective functions like  $C_{\max}$ ,  $L_{\max}$ ,  $\sum w_j C_j$  etc., further criteria may be considered, for example:

$\sum c_j^F \beta^{C_j}$	net present value ( $c^F$ cash flow, $\beta$ discount factor)
$\sum c_k f(r_k(S, t))$	resource leveling ( $c_k$ cost per unit of resource $k$ , $r_k(S, t)$ usage of resource $k$ at time $t$ given schedule $S$ )
$\sum c_k \max r_k(S, t)$	resource investment

Different types of functions  $f$  which have been considered in literature and practice will be discussed in Section 7.1.

Some of the models covered in this paper can now be classified as follows:

- $PS | prec | C_{\max}$ : This model forms the core problem among the class of resource-constrained project scheduling problems. Basically, while minimizing the project's makespan, we have to observe precedence and resource constraints. Recently, a couple of papers have contributed new solution procedures. However, the problem is still rather challenging from a computational point of view. Methods for solving this model are reviewed in Section 3.
- $MPS | prec | C_{\max}$ : Models of this class capture resource–resource and time–resource tradeoffs. Hence, they come more close to what can be observed in reality of project management. Methods for solving this model are reviewed in Section 5.
- $PS | temp | C_{\max}$ : In many applications, beside minimum time lags, maximum start–start time lags between activities must be observed. Here, already the feasibility problem is NP-complete in the strong sense. Methods for solving this model are reviewed in Section 6.
- $PS | temp | \sum c_k f(r_k(S, t))$ : In some applications the availability of renewable resources is limited and, in addition, we have to come up with a schedule which levels the resource usage over time. Methods for solving this model and related ones are reviewed in Section 7.

### 3. Single-mode case

In this section enumerative and heuristic methods for solving the basic resource-constrained project scheduling problem  $PS | prec | C_{\max}$  will be summarized. Assume that the project consists of a set  $V = \{0, 1, \dots, n, n+1\}$  of activities where activity  $j = 0$  ( $j = n+1$ ) is a fictitious beginning (termination) activity. The network is assumed to be acyclic and depicted by an activity-on-node network with nodes as activities and arcs as precedence relations. Preemption is not allowed. There are scarce renewable resources. All data are assumed to be integer-valued. The objective is to find a makespan-minimal schedule that meets the constraints imposed by the precedence relations and by limited resource availabilities.

Given an upper bound  $T$  on the minimum project duration we can use the precedence relations to derive time windows, i.e. intervals  $[EC_j, LC_j]$ , with earliest completion time  $EC_j$  and latest completion time  $LC_j$ , containing the precedence feasible completion times of activity  $j \in V$ , by forward and backward recursion. Analogously, the interval  $[ES_j, LS_j]$  bounded from below and above by the earliest start time  $ES_j$  and latest start time  $LS_j$ , respectively, can be calculated to reflect the precedence feasible start times. In general,  $PS | prec | C_{\max}$  is formulated as a 0-1 integer program which makes use of variables  $x_{jt} = 1$ , if activity  $j$  is completed in period  $t$  (0, otherwise). Alternatively, it is stated similar to what is presented in Section 6.1 for the more general  $PS | temp | C_{\max}$ . For the sake of shortness, we do not present a formal model here.

Section 3.1 describes recent branch-and-bound approaches for  $PS | prec | C_{\max}$  while Section 3.3 surveys heuristics. Lower bounds are important for both types of methods. They are the subject of Section 3.2. In Section 3.4 computational results are briefly discussed.

Within the last years branch-and-cut methods improved the solvability of several combinatorial optimization problems substantially. The generation of valid inequalities and their propagation through an LP solver might be considered as an early effective start of propagation of constraints based on consistency tests. In scheduling successful

applications have been achieved for disjunctive scheduling problems (e.g. job shop scheduling) and are currently going to be extended to  $PS | prec | C_{max}$ . The aim of such efforts is to accelerate heuristics or branch-and-bound methods through an early detection of nonattractive or infeasible nodes. A comprehensive introduction into constraint propagation has been provided by e.g. Tsang [195]. Constraint propagation-based consistency tests are described in Appendix A.

### 3.1. Branch-and-bound methods

Starting with an early work of Johnson [101] a variety of branch-and-bound algorithms have been developed for  $PS | prec | C_{max}$ . Most of them use partial schedules which are associated with the vertices of the enumeration tree. The branching process consists of extending the partial schedule in different ways. Dominance rules, lower bounds, and immediate selection allow to decrease the number of alternatives for extending the partial schedule. The methods use different branching schemes and pruning methods. In general, depth-first-search is used in order to keep memory requirements low.

*The Precedence Tree:* Patterson et al. [155] proposed an algorithm guided by the so-called precedence tree. The procedure begins with starting the dummy beginning activity at time 0. At each level  $g$  of the branch-and-bound tree, the set  $SJ_g$  of the currently scheduled activities and the set  $EJ_g$  of the eligible activities, that is, those activities the predecessors of which are already scheduled, is determined. Then an eligible activity  $j_g$  is selected. Now the earliest precedence and resource feasible start time  $S_{j_g}$  that is not less than the start time assigned on the previous level of the search tree is computed. Then we branch to the next level. If the dummy termination activity is eligible, a complete schedule has been found. In this case, backtracking to the previous level occurs. Here, the next untested eligible activity is chosen. If all eligible activities have been tested, we track another step back. Each branch from the root to a leaf of the precedence tree corresponds to a permutation of the set of activities which is precedence feasible in

the sense that each predecessor of an activity  $j_g$  has a smaller index in the sequence than  $j_g$ . Recently, this algorithm has been enhanced with powerful search tree reduction techniques by Sprecher [185].

*Delay Alternatives:* This algorithm bases on the concept of delay alternatives used by Christofides et al. [40] which has been enhanced by De-meulemeester and Herroelen [48]. In contrast to the precedence tree algorithm, here each level  $g$  of the branch-and-bound tree is associated with a fixed time instant  $t_g$  (decision point) at which activities may be started. Consequently, a different definition of eligible activities is used in this algorithm: A currently unscheduled activity  $j$  is called eligible at time  $t_g$  if all of its predecessors  $i$  are scheduled with a completion time  $C_i \leq t_g$ . Furthermore, an activity  $j$  with start time  $S_j$  is said to be in process at time  $t_g$  if we have  $S_j \leq t_g < S_j + p_j$ . The proceeding at the current level  $g$  of the branch-and-bound tree is as follows: The new decision point  $t_g$  is determined as the earliest completion time of the activities in process at  $t_{g-1}$ . Note that, due to the constant availability levels of the renewable resources, only finish times of scheduled activities need to be considered for starting unscheduled ones. Using the set  $FJ_g$  of the activities that are finished at or before the decision point, the set  $EJ_g$  of the eligible activities is computed. Having started all eligible activities by adding them to the set  $JIP_g$  of the activities in process, may have caused a resource conflict. Thus, the set of the minimal delay alternatives is computed according to the following definition: A delay alternative  $\mathcal{D}\mathcal{A}_g$  is a subset of  $JIP_g$  such that for each renewable resource  $k \in \mathcal{R}$  it is  $\sum_{j \in JIP_g \setminus \mathcal{D}\mathcal{A}_g} r_{jk} \leq R_k$ . A delay alternative  $\mathcal{D}\mathcal{A}_g$  is called minimal if no proper subset of  $\mathcal{D}\mathcal{A}_g$  is a delay alternative. A minimal delay alternative is selected and the activities to be delayed are removed from the current partial schedule. Note, if no resource conflict occurs, the only minimal delay alternative is the empty set. We store the start times of an activity  $j$  to be delayed because this information has to be restored during backtracking. Then it is branched to the next level and the next decision point is computed. If the schedule is complete now, backtracking is performed and the next minimal delay alternative is tested. Clearly,

this procedure is different from the precedence tree algorithm in that sets of activities instead of (single) activities are started at each level of the branch-and-bound tree. Moreover, here the time instant at which activities may be started is determined before the activities themselves are selected. Finally, in contrast to the precedence tree algorithm, this approach allows to withdraw scheduling decisions at the current level that have been made at a lower level.

*Extension Alternatives:* Stinson et al. [188] proposed to use extension alternatives to construct partial schedules. As in the previous algorithm, each level  $g$  of the branch-and-bound tree is associated with a decision point  $t_g$ , a set  $JIP_g$  of the activities in process, a set  $FJ_g$  of the finished activities, and a set  $EJ_g$  of eligible activities. Then the current partial schedule is extended by starting a subset of the eligible activities at the decision point without violating the resource constraints. More precisely, an extension alternative  $\mathcal{E}\mathcal{A}_g$  is a subset of the eligible set for which  $\sum_{j \in JIP_g \cup \mathcal{E}\mathcal{A}_g} r_{jk} \leq R_k$  holds for each resource  $k \in \mathcal{R}$  and, moreover,  $\mathcal{E}\mathcal{A}_g \neq \emptyset$  if  $JIP_g = \emptyset$ . Note, in order to secure that the algorithm terminates, we may only have non-empty extension alternatives if no activities are in process. However, if there are currently activities in process, the empty set is always an extension alternative which must be tested in order to guarantee optimality. At the current level  $g$  of the branch-and-bound tree the procedure is as follows: Determine the new decision point and compute the set of the eligible activities and the set of extension alternatives. Finally, select an extension alternative  $\mathcal{E}\mathcal{A}_g$  and start the corresponding activities before branching to the next level. The backtracking mechanism equals the one of the previous algorithm. Note that this procedure is different from the previous algorithm: Whereas the former includes the possibility to delay activities that have been started on a lower than the current level, the latter does not allow to withdraw a scheduling decision of a lower level. As a consequence, we may not restrict the search to “maximal” extension alternatives while we do not lose optimality when considering only minimal delay alternatives. Note, Stinson et al. [188] introduced the procedure solely by means of an example.

*Block Extensions:* Mingozzi et al. [126] consider a slightly different approach based on the following ideas. There exists an optimal schedule defining times

$$t_0 = 0 < t_1 < t_2 < \dots < t_l$$

and corresponding sets of activities  $A_1, \dots, A_l$  such that

- (i) each  $t_i (i > 0)$  is the finishing time of some activity,
- (ii) all activities in  $A_i$  can be processed jointly during  $[t_{i-1}, t_i[$  ( $i = 1, \dots, l$ ),
- (iii) if an activity  $j \in A_i$  is not finished in  $[t_{i-1}, t_i[$  it will also be processed in  $[t_i, t_{i+1}[$ , and
- (iv) all predecessors of any activity which start at time  $t_i$  are scheduled before time  $t_i$ .

A block consists of such an interval  $[t_{i-1}, t_i[$  with a set  $A_i$  of activities which can be processed jointly. Furthermore a partial schedule is defined by a sequence of blocks satisfying conditions (iii) and (iv). Then it is branched by adding new blocks providing again partial schedules.

*Schedule schemes:* The branch-and-bound algorithm developed by Brucker et al. [32] generalizes branch-and-bound methods for the job shop scheduling problem and the multiprocessor task scheduling problem (cf. [30,118]). It also uses concepts which can be found in Bartusch et al. [12]. Instead of using partial schedules, sets of feasible schedules are represented by the so-called schedule schemes. Schedule schemes can be motivated as follows.

For two arbitrary activities a schedule  $S$  induces either a parallelity relation  $i \parallel j$  or one of the two conjunctions  $i \rightarrow j$  or  $j \rightarrow i$ .  $i \rightarrow j$  holds if and only if  $i$  finishes before the start time of  $j$ .  $i \parallel j$  means that  $i$  and  $j$  are processed in parallel for at least one time unit. We get sets of schedules by relaxing these relations.  $i \rightarrow j$  or  $j \rightarrow i$  are relaxed by the disjunction  $i - j$ .  $i - j$  means that we have either  $i \rightarrow j$  or  $j \rightarrow i$ . Furthermore disjunctions  $i - j$  and parallelity relations  $i \parallel j$  can be relaxed to flexibility relations  $i \sim j$ .  $i \sim j$  means that it is undecided yet which of the two relations  $i - j$  or  $i \parallel j$  holds.  $\mathcal{C}, \mathcal{D}, \mathcal{N}$  and  $\mathcal{W}$  denote the sets of conjunctions, disjunctions, parallelity relations, and flexibility relations, respectively.  $(\mathcal{C}, \mathcal{D}, \mathcal{N}, \mathcal{W})$  is a schedule

scheme if for any two different activities  $i, j$  exactly one of the following relations holds:  $i \rightarrow j \in \mathcal{C}$  or  $j \rightarrow i \in \mathcal{C}$  or  $i - j \in \mathcal{D}$  or  $i || j \in \mathcal{N}$  or  $i \sim j \in \mathcal{U}$ . A schedule scheme  $(\mathcal{C}, \mathcal{D}, \mathcal{N}, \mathcal{U})$  defines a set of feasible schedules (which may be empty), namely all feasible schedules which satisfy all the relations in  $\mathcal{C}, \mathcal{D}, \mathcal{N}$ .

If  $\mathcal{C}_0$  is the set of all given precedence relations,  $\mathcal{D}_0$  is the set of all pairs of activities which cannot be processed in parallel due to the resource constraints, and  $\mathcal{U}_0$  the set of all remaining pairs  $i \sim j$ , then  $(\mathcal{C}_0, \mathcal{D}_0, \emptyset, \mathcal{U}_0)$  represents the set of all feasible schedules.  $(\mathcal{C}_0, \mathcal{D}_0, \emptyset, \mathcal{U}_0)$  corresponds to the root of the enumeration tree.

For a schedule scheme of the form  $(\mathcal{C}, \mathcal{D}, \mathcal{N}, \emptyset)$  it can be shown that either no feasible schedule satisfying all the relations exists or a dominating feasible schedule can be calculated (cf. [117]). Both can be done in  $O(n^3)$  time. Thus, schedule schemes of the form  $(\mathcal{C}, \mathcal{D}, \mathcal{N}, \emptyset)$  can be treated as leaves of the enumeration tree and one can branch by replacing a flexibility relation  $i \sim j$  by  $i - j$  or  $i || j$ .

The inclusion of conjunctions in schedule schemes allows to form a temporal analysis in each node of the enumeration tree. By this analysis new conjunctions, disjunctions or parallelity relations are deduced. Furthermore, for the activities time windows can be calculated which improve lower bound calculations.

*Minimal forbidden sets:* Igelmund and Radermacher [99,100] have introduced a branching scheme based on minimal forbidden sets. A more detailed discussion of these concepts can be found in Sections 6.2 and 8.2 of this survey.

Beside lower bounds which are described in Section 3.2, dominance rules are successfully used within e.g. the partial enumeration algorithms of Demeulemeester and Herroelen [48,49] and Sprecher [185] in order to prune large parts of the search tree. Among the most powerful dominance rules is the cutset rule which makes use of stored information about already evaluated partial schedules. During the search process the rule compares the current partial schedule with the stored data. If it can be proven that any solution obtainable from the current partial schedule cannot be better than a solution obtainable from a previously evaluated partial schedule the infor-

mation of which has been stored, then backtracking may be performed. A description of such rules is skipped here for the sake of shortness and the reader is referred to Section 5.2 where some rules are outlined.

### 3.2. Lower bounds

Usually lower bounds for the optimal solution value of  $PS | prec | C_{\max}$  can be calculated by relaxing some of the constraints and solving the relaxed problem to optimality. Relaxation of the resource constraints leads to the critical path length which provides a simple lower bound. Stinson et al. [188] improve this bound by adding an activity  $i$  which does not belong to a critical path  $CP$ . They calculate a maximal number  $e_i$  of time units activity  $i$  can be processed in parallel with  $CP$ . By adding  $\max\{0, p_i - e_i\}$  to the critical path length a new lower bound is provided. Instead of adding a single activity, Demeulemeester and Herroelen augment a critical path  $CP$  by a path  $P$  node-disjoint with  $CP$  and calculate a lower bound for  $CP \cup P$  using a dynamic programming procedure. In general this lower bound does not coincide with the optimal solution value for  $CP \cup P$  (cf. [172]). However, the two-path relaxation can be solved to optimality by a graphical method developed for the job shop problem with two jobs. This approach also allows to find an optimal solution for  $CP \cup P$  which respects time windows for the activities in  $CP \cup P$  (cf. [28]).

A bound of Mingozzi et al. [126] is based on the following linear program which relaxes partially the precedence constraints and allows preemption. They consider maximal sets of activities which can be processed in parallel. Let  $a_1, a_2, \dots, a_q$  be the characteristic vectors of all these sets. Then the linear programming relaxation has the form

$$\min \sum_{j=1}^q x_j \quad (1)$$

$$\text{s.t.} \quad \sum_{j=1}^q a_{ij} x_j \geq p_i, \quad i = 1, \dots, n, \quad (2)$$

$$x_j \geq 0, \quad j = 1, \dots, q, \quad (3)$$

where  $x_j$  denotes the number of time units all activities represented by  $a_j$  are processed jointly. The integer version of the dual of Eqs. (1)–(3) is a set packing problem. Mingozzi et al. [126] provide lower bounds for the resource-constrained project scheduling problem by solving the set packing problem heuristically.

Baar et al. [5] solve the linear program (1)–(3) directly by applying column generation techniques. Although the number of columns is growing exponentially with the number of activities the method is quite fast. Brucker and Knust [31] enhanced the approach by taking into account time windows for the activities. These time windows are derived from the precedence constraints using a fictitious upper bound  $T$  for the makespan. Now the columns correspond to sets of activities which can be processed jointly in a given time window. The objective is to find a preemptive schedule respecting all times windows. If such a schedule does not exist,  $T$  is a lower bound. Binary search provides largest  $T$  with this property. These “destructive improvement” technique has also been used by Klein and Scholl [105] who tested various methods for proving infeasibility of an upper bound  $T$ .

Other linear programming based bounds have been introduced in connection with the branch-and-bound algorithm of Christofides et al. [40].

### 3.3. Heuristic methods

The first heuristic methods were priority-rule based scheduling methods (cf. [103]). Up to now, a multitude of priority rules were proposed and tested experimentally (cf. [4,22,42,153,194]). Priority-based heuristics have the advantage of being intuitive, easy to implement, and fast in terms of computational effort. However, they do not excel with respect to the average deviation from the optimal objective function value. Hence, recent research interests shifted to more elaborate heuristics like truncated branch-and-bound (cf. [4]), integer programming based heuristics (cf. [148]), disjunctive arc concepts (cf. [4,14]), local constraint-based analysis (cf. [196]), sampling tech-

niques (cf. [110]), and local search techniques (cf. [5,24,85,108,119,169]).

### 3.4. Computational results

Demeulemeester and Herroelen [48] tested their branch-and-bound algorithm on the Patterson-set (cf. [154]) which consists of 110 test problems with up to 51 activities. They solved these problems with an average computation time of 0.21 s (IBM PS/2 Model 70 A21, 25 MHz) outperforming the algorithm of Stinson et al. [188] by a factor of nearly 12.

Kolisch et al. [116] developed the parameter-driven project generator ProGen which thereafter has been widely used as a tool for the evaluation of algorithms proposed for resource-constrained project scheduling. Meanwhile test sets with 30, 60, 90, and 120 activities have been generated each consisting of 480 instances of various types. The 30 activity test set was also used to test the approach by Demeulemeester and Herroelen [48] on a personal computer (IBM PS/2 Model 55SX, 386SX, 15 MHz). Whereas the Patterson-set has been solved within 1.06 s on average, only 415 of the 480 problems instances have been solved within a time limit of 1000 s per problem.

Mingozzi et al. [126] report that their linear programming formulation based bounds perform better than the critical sequence bound introduced by Stinson et al. [188], and that their algorithm is competitive to the procedure presented by Demeulemeester and Herroelen [48], the best one known up to then. Demeulemeester and Herroelen [49] enhanced their approach by adapting a lower bound of Mingozzi et al. [126] and by representing four resources of 8 bit size through one 32 bit unsigned integer. Allowing as much as 24 Mbyte they could solve the entire 30 activity benchmark-set for the first time. The CPU-time averages about 34 s on a personal computer (80486, 25 MHz). Sprecher [185] compared his branch-and-bound algorithm with the enhanced version of the algorithm of Demeulemeester and Herroelen pointing out the tradeoff between computation times and memory requirements. While the enhanced algorithm of Demeulemeester and Herroelen [49] solves 479 of the 480 benchmark problems with 30

activities within average time of 12.33 s using 24 Mbyte memory, the algorithm of Sprecher solves the same problems with average time of 12.85 s using 400 Kbyte memory.

The branch-and-bound algorithm of Brucker et al. [32] solves less problems than other algorithms, but its main advantage is smaller memory requirement. For problems with up to 90 activities it uses at most 10 Mbyte. 326 of the 480 benchmark problems with 60 activities were verified for the first time within a 1 h time limit on a SUN/Sparc 20/801 workstation.

Comparing the heuristic methods it can be concluded that the adaptive search algorithm of Kolisch and Drexl [110], the tabu search method based on schedule schemes of Baar et al. [5], the genetic algorithm of Hartmann [85] and the simulated annealing algorithm of Bouleimen and Lecocq [24] provide the best results. Recently, Kolisch and Hartmann [112] performed a comparison of most of the available heuristic algorithms for larger instances. According to these results, the genetic algorithm of Hartmann [85] and the simulated annealing algorithm of Bouleimen and Lecocq [24] are the most promising candidates.

#### 4. Time–cost tradeoff problems

So far, we have only considered project networks with *fixed* (and known) processing times. A generalization of this setting that is still deterministic (and assumes complete information) is obtained by permitting processing times to *vary* according to how much the planner is willing to pay for it. In view of the next section, this control on the processing times can be interpreted as allocation of a nonrenewable resource to the activities, where a larger allocation to an activity (i.e., a higher cost input) reduces its processing time.

The planner then aims at either minimizing the project makespan subject to a fixed upper bound on the nonrenewable resource (the *budget problem*), or at minimizing the total allocation subject to a given bound on the makespan (the *deadline problem*). As the allocation is usually measured in money, these problems are commonly referred to as time–cost tradeoff problems.

##### 4.1. Model

A formal model for time–cost tradeoff problems consists of a set  $V = \{0, 1, \dots, n, n+1\}$  of activities, a directed graph  $G = (V, E)$  of precedence constraints among the activities, and, for each activity  $j$ , a set  $\mathcal{M}_j$  of possible processing times  $p_j \in \mathcal{M}_j$ , together with a nonincreasing function  $c_j: \mathcal{M}_j \rightarrow \mathbb{R}_+$  that models the individual tradeoff between processing time  $p_j$  of activity  $j$  and the cost (or amount of resource)  $c_j(p_j)$  allocated to it. Section 5 covers the model  $MPS | prec | C_{\max}$ , a generalization of the model dealt with in this section. There,  $\mathcal{M}_j$  will be used in a similar way to denote the set of modes.

Different assumptions on the sets  $\mathcal{M}_j$  and the cost functions  $c_j$  lead to different subcases of time–cost tradeoff problems.

For instance, if every  $\mathcal{M}_j$  is a closed interval  $\mathcal{M}_j = [a_j, b_j]$  and  $c_j$  is affine linear and decreasing on  $\mathcal{M}_j$ , we have the *linear time–cost tradeoff problem* introduced by Kelley and Walker [104].

If, on the other hand, every  $\mathcal{M}_j$  is a discrete (i.e., finite) set and  $c_j$  is decreasing on  $\mathcal{M}_j$ , we have the *discrete time–cost tradeoff problem* introduced by Harvey and Patterson [87] and Hindelang and Muth [96].

A realization  $p \in \mathbb{R}_+^{n+2}$  of the project is an assignment of processing times  $p_j \in \mathcal{M}_j$  to activities  $j \in V$ . The *total cost*  $c(p)$  of the realization  $p$  is given by  $c(p) = \sum_{j \in V} c_j(p_j)$ . The makespan  $C_{\max}(p)$  of the realization  $p$  is the makespan of the earliest start schedule of the project when activity  $j$  has processing time  $p_j$ , i.e., the length of a longest path in  $G = (V, E)$  with  $p_j$  as “length” of vertex  $j \in V$ .

Fixing either cost or time, we obtain two related optimization problems with the objective to minimize the other parameter.

*Budget problem:* For a given nonnegative budget  $b \geq 0$ , find a realization  $p$  with  $c(p) \leq b$  that minimizes the makespan  $C_{\max}(p)$ .

*Deadline problem:* For a given deadline  $\bar{d}$  on the makespan, find a realization  $p$  with  $C_{\max}(p) \leq \bar{d}$  that minimizes the total cost  $c(p)$ .

Usually, one wants to solve these problems for all possible budgets or deadlines. This leads to the function



$$T_{\text{opt}}(b) := \min\{C_{\text{max}}(p) \mid p_j \in \mathcal{M}_j, c(p) \leq b\}$$

giving the minimum makespan as a function of the budget  $b$ , and the function

$$B_{\text{opt}}(\bar{d}) := \min\{c(p) \mid p_j \in \mathcal{M}_j, C_{\text{max}}(p) \leq \bar{d}\},$$

giving the minimum cost as a function of the deadline  $\bar{d}$  (the *project cost curve*).

The budget problem is a special case of *MPS1 | prec | C<sub>max</sub>* which is covered in Section 5. While in this section we have no renewable resources and only one single nonrenewable resource, Section 5 considers the general multi-mode version. Similarly, the deadline problem combines characteristics of the multi-mode case (Section 5) and of the resource levelling problem (Section 7). In our notation, it could be denoted by *MPS1 | prec |  $\sum c_k r_k(S, t)$* .

#### 4.2. Exact algorithms

Kelley and Walker [104] discuss both the budget problem and the deadline problem within the context of the linear time–cost tradeoff problem. In this setting (and also in a more general one, see [15]), the project cost curve  $B_{\text{opt}}$  can be obtained as the inverse function of  $T_{\text{opt}}$ , so it suffices to consider only  $B_{\text{opt}}$ .

For every fixed deadline  $\bar{d}$ ,  $B_{\text{opt}}(\bar{d})$  can be expressed as a special linear program whose dual resembles a min-cost flow problem (see [74]). Using standard results from parametric linear optimization it hence follows that  $B_{\text{opt}}(\bar{d})$  is a piecewise linear and convex function of the parameter  $\bar{d}$ .

Fulkerson [74] and Kelley [102] independently developed the same algorithm to compute the project cost curve  $B_{\text{opt}}$ . This algorithm uses an activity-on-arc representation of the network and iteratively calculates a sequence of less and less “cheap” cuts in the current network of critical activities by which the makespan is reduced. Every breakpoint of the project cost curve corresponds to a change of the current cut to a more expensive one.

Every such cut can be determined by a max-flow computation in which the capacities are de-

rived from the slopes of the linear cost functions  $c_j$  of the critical activities. These ideas have subsequently been improved by Phillips and Dessouky [159,160].

This algorithm is polynomial per cut ( $O(|V|^2 \log|V|)$ ) by standard flow methods (cf. [79]), but the number of cuts to be computed may be large. Skutella [178] provides a class of examples for which the project cost curve has exponentially many breakpoints, thus requiring an exponential number of cut calculations.

The case where the possible processing times are discrete is quite common in practice. Only recently, De et al. [46] showed that, given the budget  $b$ , it is strongly NP-complete to decide whether there is a realization  $p$  such that  $c(p) \leq b$  and  $C_{\text{max}}(p) \leq 2$ . This holds already for activities with at most two processing time alternatives, i.e.,  $|\mathcal{M}_j| \leq 2$ .

Due to the practical importance of the problem, many (exponential time) exact algorithms have been proposed. Early examples are dynamic programming approaches by Hindelang and Muth [96] and Robinson [164], and an enumeration algorithm by Harvey and Patterson [87].

The currently best known algorithms still rely on dynamic programming, but exploit in addition the decomposition structure of the underlying network. The decomposition that facilitates the computation is known as *modular decomposition* or *substitution decomposition* and has many applications in network and other combinatorial optimization problems, see the comprehensive article by Möhring and Radermacher [129].

Its usefulness for the time–cost tradeoff problem was first observed by Frank et al. [71] and Rothfarb et al. [165] for the special case of series–parallel decompositions. The general decomposition theorem that involves arbitrary modules is due to Billstein and Radermacher [15] (see also [130]).

Because of the modular decomposition, the project cost curve needs only to be evaluated for certain indecomposable subnetworks (the factors in a composition series) of the original network. This is done by “transforming” such an indecomposable network to a series–parallel network

and then performing the “easy” calculations for the series–parallel case. The transformation into a series–parallel network successively identifies certain nodes for “duplication”. Any such duplication transforms the network “closer” to a series–parallel one, but increases the computation time by a multiplicative factor.

This idea seems to be due to Robinson [164] and has been further developed by Bein et al. [13], De et al. [45] and Elmaghraby [66]. The same ideas also came up in reliability theory and seem to have influenced each other, see [6].

Demeulemeester et al. [50] provide the first implementation of this approach. They implement two strategies for finding the nodes for duplication. The first follows the theory of Bein et al. [13], which results in the minimum number of duplications required, while the second tries to minimize the number of realizations that have to be considered during the algorithm. Demeulemeester et al. [50] report on computational experience for networks with up to 45 activities without identifying a clear winner between the two strategies.

The crucial parameter in the theoretical run-time analysis of this algorithmic approach is the minimum number of node duplications needed to transform an activity-on-arc network into a series–parallel network. Bein et al. [13] refer to it as the *reduction complexity* of the network. It provides a measure for the “distance” of the given network from being series–parallel.

Such a distance measure is important for the design of polynomial-time algorithms for many network problems (see also Section 8.1 on stochastic scheduling), since computational approaches for series–parallel graphs can often be extended to algorithms for arbitrary graphs that are exponential only in the “distance” from being series–parallel, rather than in its size.

Another measure for this distance is the *factoring complexity* also introduced by Bein et al. [13], which is based on a special way of describing all paths from the source to the sink of the network. Bein et al. [13] showed that the factoring complexity provides an upper bound for the reduction complexity. Naumann [137] showed that both measures are in fact equal.

### 4.3. Approximation algorithms

The approximation behavior of the discrete time–cost tradeoff problem has recently been analyzed by Skutella [177].

He first presents a polynomial reduction to the case where every activity has at most two processing times, and one of them is zero. So  $\mathcal{M}_j = \{0, \bar{p}_j\}$  or  $\mathcal{M}_j = \{\bar{p}_j\}$  for activity  $j \in V$ .

For such a discrete time–cost tradeoff problem, Skutella defines a natural linear relaxation, which replaces  $\mathcal{M}_j$  by the interval  $\tilde{\mathcal{M}}_j = [0, \bar{p}_j]$  and takes as cost function  $c_j$  the linear interpolation between  $c_j(0)$  and  $c_j(\bar{p}_j)$ . All other parameters remain the same.

Now consider the deadline problem  $P$  for a fixed deadline  $\bar{d}$ . One then first solves the linear relaxation  $\tilde{P}$  for the same deadline  $\bar{d}$ , which yields an optimal realization  $\tilde{p}$  of  $\tilde{P}$  in polynomial time. Since all parameters are assumed to be integral, the obtained optimal realization  $p$  will in this special case also be integral, but  $\tilde{p}_j$  need not be in  $\{0, \bar{p}_j\}$ . This solution  $\tilde{p}$  is then “rounded” to a solution of the original problem  $P$  by rounding the processing time  $\tilde{p}_j$  of those activities  $j$  with  $0 < \tilde{p}_j < \bar{p}_j$  to the lower value  $p_j = 0$ . Rounding to the lower value is necessary in order to preserve the deadline  $\bar{d}$ .

If the budget problem is considered, the rounding must be done into the other direction, i.e.,  $p_j = \bar{p}_j$ , thus preserving the budget condition. In both cases, the produced realization  $p$  of the discrete time–cost tradeoff problem is an  $\ell$ -approximation, where  $\ell$  is the largest occurring processing time of any activity. So  $C_{\max}(p) \leq \ell \cdot T_{\text{opt}}(b)$  and  $c(p) \leq \ell \cdot B_{\text{opt}}(\bar{d})$ , respectively.

For the deadline problem, the performance guarantee of  $\ell$  cannot be improved by this rounding algorithm and it is open whether it can be improved at all. For the budget problem, however, Skutella develops better approximation algorithms. Unlike the situation for the deadline problem, he can now repair a budget violation by rounding some processing times to the higher value  $\bar{p}_j$ , thus “saving” part of the budget that can then be “reinvested” to shorten “critical” activities to the lower value 0.

For projects with  $\bar{p}_j \in \{0, 1, 2\}$  this leads to a  $\frac{3}{2}$ -approximation, i.e., for a given budget  $b$ , the al-

gorithm produces a realization  $p$  with  $c(p) \leq b$  and  $C_{\max}(p) \leq \lceil \frac{3}{2} T_{\text{opt}}(b) \rceil$ . The running time of the algorithm is  $O(|V|^3 \log |V|)$ .

The NP-completeness of deciding whether a makespan of 2 can be realized with a given budget shows that the performance ratio of  $\frac{3}{2}$  cannot be improved (unless  $\mathcal{P} = \mathcal{NP}$ ).

For projects with  $p_j \leq \ell$ , Skutella uses additional partitioning techniques and obtains a strongly polynomial approximation algorithm with a performance guarantee of  $2(\log_2 \ell + 1)$ . A different variant yields  $\frac{3}{2} \log_2 \ell + 3$ .

Another idea for approximation algorithms consists in relaxing also the tight constraint (budget or deadline), thus arriving at the so-called *bicriteria approximation algorithms*. Using ideas similar to those above, Skutella shows that, for a value  $0 < \mu < 1$  and an optimal time–cost pair  $(\bar{d}, b)$  (i.e.,  $\bar{d} = T_{\text{opt}}(b)$  and  $b = B_{\text{opt}}(\bar{d})$ ), one can in polynomial time construct a realization  $p$  such that  $c(p) < (1/(1 - \mu))b$  and  $C_{\max}(p) \leq (1/\mu)\bar{d}$ . For  $\mu = \frac{1}{2}$ , this yields a realization which is at most twice as expensive and twice as long as an optimal realization for the given deadline or budget. Choosing  $\mu$  uniformly at random in the interval  $[1/e, 1]$  one obtains improved approximation ratios of  $e/(e - 1) \approx 1.58$  for the expected cost and expected makespan.

## 5. Multi-mode case

This section covers exact and heuristic algorithms for solving  $MPS \mid prec \mid C_{\max}$  which is defined as follows. Like in the previous sections the project network is assumed to be acyclic and topologically sorted. Each activity  $j \in V$  may be executed in one out of a set of  $\mathcal{M}_j$  modes. Also, each activity may not be preempted and a mode once selected may not be changed, that is, an activity  $j$  once started in mode  $m \in \mathcal{M}_j$  has to be completed in mode  $m$  without interruption. Processing activity  $j$  in mode  $m$  takes  $p_{jm}$  periods and is supported by a set  $\mathcal{R}^\rho$  of renewable and a set  $\mathcal{R}^v$  of nonrenewable resources. Considering a horizon  $T$ , that is, an upper bound on the project’s makespan, we have an available amount of  $R_k^\rho$  units of renewable resource  $k$  in period  $t = 1, \dots, T$  like in

Section 3. The overall capacity of the nonrenewable resource  $k \in \mathcal{R}^v$  is given by  $R_k^v$ . If activity  $j$  is processed in mode  $m$  then  $r_{jkm}^\rho$  units of the renewable resource  $k$  are used each period activity  $j$  is in process. Similarly, activity  $j$  consumes  $r_{jkm}^v$  units of the nonrenewable resource  $k$ . In general, the parameters are assumed to be integer-valued. We assume the modes to be labeled with respect to nondecreasing processing times, that is,  $p_{jm} \leq p_{j,m+1}$  for all activities  $j \in V$  and modes  $m \in \{1, \dots, |\mathcal{M}_j| - 1\}$ . The objective is to find a makespan-minimal schedule  $\mathcal{S}$  that meets the constraints imposed by the precedence relations and by limited resource availabilities. Similar to  $PS \mid prec \mid C_{\max}$  in general also  $MPS \mid prec \mid C_{\max}$  is formulated mathematically in terms of a binary optimization model which makes use of binary variables  $x_{jmt} = 1$ , if activity  $j$  is completed in mode  $m$  in period  $t$  (0, otherwise). For the sake of shortness, we do not present a formal model here.

If  $|\mathcal{R}^v| \geq 2$  and  $|\mathcal{M}_j| \geq 2, j \in V$ , then finding a feasible solution is NP-complete (cf. [109]). However, presuming feasibility and a constant per-period availability of the renewable resources, an upper bound on the minimum makespan  $T$  is given by the sum of the maximum activity processing times. Given an upper bound  $T$  we can use the precedence relations and the modes of the shortest processing times to derive time windows, i.e. intervals  $[EC_j, LC_j]$  similar to what has been explained in Section 3.

### 5.1. Exact algorithms

Optimal procedures for solving  $MPS \mid prec \mid C_{\max}$  generalize procedures described in Section 3.1 for solving the special case  $PS \mid prec \mid C_{\max}$ . We summarize three algorithms.

*The Precedence Tree:* Sprecher and Drexler [186] improved the precedence tree algorithm introduced by Patterson et al. [155] by including new bounding criteria. Here, on level  $g$  of the branch-and-bound tree an eligible activity  $j_g$  and, subsequently, a mode  $m_{j_g}$  of this activity are selected. Each combination of an eligible activity and a related mode corresponds to a descendant of the current node in the branch-and-bound tree.

*Mode and delay alternatives:* We summarize the branch-and-bound approach proposed by Sprecher et al. [187]. An eligible activity  $j$  scheduled in mode  $m_j$  with start time  $S_j$  is said to be in process at time  $t_g$  if we have  $S_j \leq t_g < S_j + p_{jm_j}$ . Eligible activities are (temporarily) started at the decision point that have already been assigned a mode at a previous level of the search tree. If there are eligible activities that have not yet been assigned a mode, that is, if  $EJ_g \setminus EJ_{g-1}$  is not empty, then the set of mode alternatives is computed: A mode alternative is a mapping which assigns each activity  $j \in EJ_g \setminus EJ_{g-1}$  a mode  $m_j \in \mathcal{M}_j$ . Selecting a mode alternative, the remaining eligible activities can be (temporarily) started at the decision point as well. Having started all eligible activities by adding them to the set  $JIP_g$  of the activities in process, may have caused a resource conflict. Thus, the set of the minimal delay alternatives is computed according to the following definition: A delay alternative  $\mathcal{D}\mathcal{A}_g$  is a subset of  $JIP_g$  such that for each renewable resource  $k \in \mathcal{R}^\rho$  it is  $\sum_{j \in JIP_g \setminus \mathcal{D}\mathcal{A}_g} r_{jkm_j}^\rho \leq R_k^\rho$ . Observe that each combination of a mode alternative and a related minimal delay alternative corresponds to a descendant of the current node in the branch-and-bound tree.

*Mode and extension alternatives:* Using again the concept of mode alternatives extension alternatives are introduced by Hartmann and Drexel [86] to construct partial schedules. More precisely, an extension alternative  $\mathcal{E}\mathcal{A}_g$  is a subset of the eligible set for which  $\sum_{j \in JIP_g \cup \mathcal{E}\mathcal{A}_g} r_{jkm_j}^\rho \leq R_k^\rho$  holds for each renewable resource  $k \in \mathcal{R}^\rho$  and, moreover,  $\mathcal{E}\mathcal{A}_g \neq \emptyset$  if  $JIP_g = \emptyset$ . At level  $g$  of the branch-and-bound tree we determine the new decision point and the set of the eligible activities. Then we compute the set of mode alternatives for fixing the modes of the eligible activities that have not been eligible before, that is, those activities the modes of which have not yet been fixed. After selecting a mode alternative, compute the set of extension alternatives. Finally, select an extension alternative  $\mathcal{E}\mathcal{A}_g$  and start the corresponding activities before branching to the next level. Each combination of a mode alternative and a related extension alternative corresponds to a descendant of the current node in the branch-and-bound tree.

Recently, the more general problem  $MPS | temp | C_{\max}$  with general temporal constraints given by minimum and maximum start–start time lags between activities has been the subject of research in Heilmann [88], where an exact branch-and-bound procedure is presented.

A combination of the discrete time–cost trade-off problem covered in Section 4 and of the multi-mode case dealt with in this section has been studied by Ahn and Erengüç [1].

## 5.2. Dominance rules

In Hartmann and Drexel [86] a description of several bounding rules can be found. Some of them will be revisited in what follows.

*Non-delayability rule:* If an eligible activity cannot be feasibly scheduled in any mode in the current partial schedule without exceeding its latest finish time, then no other eligible activity needs to be examined on this level.

*Local left shift rule:* If an activity that has been started at the current level of the branch-and-bound tree can be locally left shifted without changing its mode, then the current partial schedule needs not be completed.

*Multi-mode rule:* Assume that no currently unscheduled activity will be started before the finish time of a scheduled activity  $j$  when the current partial schedule is completed. If a multi-mode left shift or a mode reduction of activity  $j$  with resulting mode  $m'_j$ ,  $1 \leq m'_j \leq |M_j|$ , can be performed on the current partial schedule and, moreover, if  $r_{jkm'_j}^v \leq r_{jkm_j}^v$  holds for each nonrenewable resource  $k$ , then the current partial schedule need not be completed.

*Order swap rule:* Consider a scheduled activity the finish time of which is less than or equal to any start time that may be assigned when completing the current partial schedule. If an order swap on this activity together with any of those activities that finish at its start time can be performed, then the current partial schedule need not be completed.

*Cutset rule:* Defining a cutset of a partial schedule  $\mathcal{P}\mathcal{S}$  as the set of the activities scheduled in  $\mathcal{P}\mathcal{S}$ , Sprecher and Drexel [186] proposed the following rule. Let  $\overline{\mathcal{P}\mathcal{S}}$  denote a previously eval-

uated partial schedule with cutset  $CS(\overline{\mathcal{PS}})$ , maximal finish time  $f^{\max}(\overline{\mathcal{PS}})$  and leftover capacities  $R_k^v(\overline{\mathcal{PS}})$  of the nonrenewable resources  $k$ . Let  $\mathcal{PS}$  be the current partial schedule considered to be extended by scheduling some activity  $j$  with start time  $S_j$ . If we have  $CS(\mathcal{PS}) = CS(\overline{\mathcal{PS}})$ ,  $S_j \geq f^{\max}(\overline{\mathcal{PS}})$  and  $R_k^v(\mathcal{PS}) \leq R_k^v(\overline{\mathcal{PS}})$  for all  $k \in \mathcal{R}^v$ , then  $\mathcal{PS}$  needs not be completed.

*Immediate selection:* Consider an eligible activity  $j$  no mode of which is simultaneously performable with any currently unscheduled activity in any mode. If the earliest feasible start time of each other eligible activity in any mode is equal to the maximal finish time of the currently scheduled activities, then  $j$  is the only eligible activity that needs to be selected for being scheduled on the current level of the branch-and-bound tree.

### 5.3. Heuristic algorithms

Heuristic algorithms for solving  $MPS | prec | C_{\max}$  have for instance been provided by Drexel [60], Drexel and Grünwald [61], Özdamar [149] and Kolisch and Drexel [111]. Slowiński et al. [179] address the same set of constraints, but attack the multi-criteria version of the problem.  $MPSm, \sigma, \rho; 0 | prec | C_{\max}$  is the subject of Boctor [23]. While Boctor, Drexel, and Drexel and Grünwald analyze priority rule based multi-pass heuristics, Slowiński et al. provide simulated annealing algorithms, Özdamar favors a genetic algorithm and Kolisch and Drexel present problem specific local search algorithms.

Recently, Hartmann [84] developed the most effective and efficient heuristic algorithm for solving the general version of the problem dealt with in this section. It is a generalized version of the genetic algorithm already mentioned in Section 3.3 and basically works as follows. The genetic algorithm generates an initial population, i.e. the first generation, containing  $POP$  individuals and then determines their fitness values.  $POP$  is assumed to be an even integer. Then the population is randomly partitioned into pairs of individuals. To each pair of (parent) individuals, the crossover operator produces two new offsprings. Subsequently, the mutation operator is applied to the

genotypes of the newly produced children. After computing the fitness of the offsprings, they are added to the current population, leading to a population size of  $2 \cdot POP$ . Then the selection operator is applied to reduce the population to its former size  $POP$  and to obtain the next generation to which again the crossover operator is applied. This process is repeated for a prespecified number of generations which is denoted as  $GEN$ .

Now a short description of the genetic operators crossover, mutation, selection is given (for details the reader is referred to Hartmann [84]).

Consider two individuals selected for *crossover*, a mother and a father. Then two random integers  $w_1$  and  $w_2$  with  $1 \leq w_1, w_2 \leq n$  are drawn. Now two new individuals, a daughter and a son, are produced from the parents. The daughter is defined as follows: In the sequence of activities of the daughter, the positions  $i = 1, \dots, w_1$  are taken from the mother. The activity sequence of positions  $i = w_1 + 1, \dots, n$  is taken from the father. However, the activities that have already been taken from the mother may not be considered again. This definition ensures that the relative positions in the parents' activity sequences are preserved. Observe that the resulting activity sequence is precedence feasible. The modes of the activities on the positions  $i = 1, \dots, w_2$  in the daughter are defined by the mother's mode assignment. The modes of the remaining activities on the positions  $i = w_2 + 1, \dots, n$  are derived from the father's mode assignment. The son is computed similarly. However, the positions  $1, \dots, w_1$  of the son's activity sequence are taken from the father and the remaining positions are determined by the mother. Analogously, the first part up to position  $w_2$  of the mode assignment of the son is taken from the father while the second part is derived from the mother. Given an activity sequence and a mode assignment for all activities an earliest start schedule is constructed.

The *mutation* is applied to each newly generated child individual and is defined as follows: Given an individual  $I$  of the current population, then two random integers  $q_1$  and  $q_2$  with  $1 \leq q_1 < n$  and  $1 \leq q_2 \leq n$  are drawn.  $q_1$  is used to modify the activity sequence by exchanging activities  $j_{q_1}^I$  and  $j_{q_1+1}^I$  if the result is an activity sequence which

fulfills the precedence constraints. Note that each of the changed activities keeps its assigned mode, that is, this modification does not change the mode assignment. Then a new mode for the activity on position  $q_2$  is randomly chosen, that is, we re-determine  $m^l(j_{q_2}^l)$  by drawing a random integer out of  $\{1, \dots, \mathcal{M}_{j_{q_2}}\}$ . While the first step may create activity sequences that could not have been produced by the crossover operator, the second step may introduce a mode that has not occurred in the current population. It should be noted that performing a mutation on an individual does not necessarily change the related schedule. This is due to the redundancy in the genetic representation.

Two variants of the *selection operator* have been considered. The first variant is a simple survival-of-the-fittest method: The original population size is restored by keeping the *POP* best individuals and removing the remaining ones from the population (ties are broken arbitrarily). The second variant is a randomized version of the survival-of-the-fittest technique.

A number of *ISL* islands are considered on which the artificial evolution as described above takes place. On each island, the evolution starts with an independently generated initial population. Let the island currently under consideration be denoted as  $i$  with  $1 \leq i < ISL$ , and let the current generation be denoted as  $g$  with  $1 \leq g \leq GEN$ . A prespecified migration probability  $w_{\text{migration}}$  is used and a random number  $q \in [0, 1]$  is drawn to control the migration between the islands: If  $q \leq w_{\text{migration}}$ , then the fittest individual of generation  $g$  leaves island  $i$  and migrates to island  $i + 1$  where it is added to the population of generation  $g$ .

The stopping criterion is either to reach a pre-specified number of islands as described above or, alternatively, to meet a given limit on the CPU time without bounding the number of islands. In the latter case, if *GEN* generations have been completed and the time limit has not yet been met, we skip to the next island and start a new evolution. Clearly, if the number of islands is given by *ISL*, at most  $ISL \cdot POP \cdot GEN$  different individuals are calculated.

The genetic algorithm is augmented by a problem specific local search method to improve

the schedule related to an individual. The approach is based on the definition of a multi-mode left shift which has been introduced by Sprecher et al. [187] in order to accelerate their branch-and-bound algorithm outlined above. A multi-mode left shift of an activity  $j$  is an operation on a given schedule which reduces the finish time of activity  $j$  without changing the modes or finish times of the other activities and without violating the constraints. Thereby, the mode of activity  $j$  may be changed.

#### 5.4. Computational results

A set of test problems constructed by the project generator ProGen which has been developed by Kolisch et al. [116] has been used. They are available in the project scheduling problem library PSPLIB. For detailed information the reader is referred to Kolisch and Sprecher [115] (cf. [114] also). The multi-mode problem sets containing instances with 10, 12, 14, and 16 nondummy activities have been used. Each of the real activities may be performed in one out of three modes. The duration of a mode varies between 1 and 10 periods. There are two renewable and two nonrenewable resources. For each problem size, a set of instances was generated by systematically varying four parameters, that is, the resource factor and the resource strength of each resource category.

In Hartmann and Drexel [86] a computational comparison of the three branching schemes in combination with bounding rules can be found. The precedence tree algorithm with the cutset rule is the fastest procedure on the average. It is two times faster than the algorithm based on mode and delay alternatives when 10 activities are considered and seven times faster for projects with 16 activities, that is, the comparison factor increases with an increasing number of activities. The algorithm based on mode and delay alternatives is at most 1.4 times faster than the algorithm based on mode and extension alternatives, hence, the latter one is outperformed by the other two algorithms with respect to average computation times. This seems to be due to the fact that branching may not be restricted to “maximal” extension alternatives. The precedence

tree algorithm is faster than the other two procedures even if the cutset rule is not included.

Noteworthy to mention that the precedence tree algorithm is more general than the other branching schemes in the sense that the case of time-varying availability profile of renewable resources can be covered.

The genetic algorithm of Hartmann [84] has been compared with the algorithm of Kolisch and Drexel [111] and Özdamar [149] on the ProGen set with 10 nondummy activities. Hartmann’s algorithm produces an average deviation of 0.22% from the optimal makespan. The procedure of Kolisch and Drexel produces an average deviation of more than 0.8% from the optimal makespan. Also, the algorithm of Özdamar has an average deviation of more than 0.8%. Hence, the average deviation produced by the genetic algorithm of Hartmann [84] is nearly four times lower than those of the two heuristics from the literature.

### 6. Minimum and maximum time lags

This section is concerned with the problem  $PS | temp | C_{max}$ , that is, maximum time lags between the start of different activities occur in addition to minimum ones. Maximum time lags are often needed in practice, for example, if simultaneous or nondelay execution of several activities is required, deadlines for subprojects or individual activities are prescribed, time windows for resources are given, or in scheduling of make-to-order production (cf. [138,139]).

Section 6.1 deals with modeling problem  $PS | temp | C_{max}$ . Section 6.2 describes branch-and-bound methods for  $PS | temp | C_{max}$ . Heuristic procedures are briefly discussed in Section 6.3. The latter two sections also summarize computational results.

#### 6.1. Model

As in Section 3,  $V = \{0, 1, \dots, n, n + 1\}$  is the set of activities of the project, which coincides with the node set of the corresponding activity-on-node project network. The fictitious activities 0 and

$n + 1$  represent the beginning and termination of the project, respectively. If there is a given minimum time lag  $d_{ij}^{min} \in \mathbb{Z}_{\geq 0}$  between the start of two different activities  $i$  and  $j$ , that is,  $S_j - S_i \geq d_{ij}^{min}$ , we introduce an arc  $(i, j)$  in the project network with weight  $\delta_{ij} = d_{ij}^{min}$ . If there is a given maximum time lag  $d_{ij}^{max} \in \mathbb{Z}_{\geq 0}$  between the start of activities  $i$  and  $j$ , that is,  $S_j - S_i \leq d_{ij}^{max}$ , we introduce an arc  $(j, i)$  with weight  $\delta_{ji} = -d_{ij}^{max}$ . The resulting network with node set  $V$ , arc set  $E$ , and arc weights  $\delta_{ij}$  which satisfy the constraints  $S_j - S_i \geq \delta_{ij}$  for  $(i, j) \in E$  generally contains cycles due to maximum time lags. An appropriate specification of the minimum and maximum time lags ensures the unique assignment of the network to the underlying project (see [139]).

Given a schedule  $S = (S_0, S_1, \dots, S_{n+1})$ ,

$$\mathcal{A}(S, t) = \{j \in V \mid S_j \leq t < S_j + p_j\}$$

is the set of activities in progress at time  $t \in \mathbb{Z}_{\geq 0}$  (or in time interval  $[t, t + 1[$  or period  $t + 1$ , respectively) and

$$r_k(S, t) = \sum_{j \in \mathcal{A}(S, t)} r_{jk}$$

is the usage of renewable resource  $k \in \mathcal{R}$  at time  $t$ . Problem  $PS | temp | C_{max}$  can then be stated as follows:

$$\min S_{n+1} \tag{4}$$

$$\text{s.t. } S_j - S_i \geq \delta_{ij} \quad (i, j) \in E$$

$$S_j \geq 0, \quad j \in V. \tag{5}$$

$$r_k(S, t) \leq R_k, \quad k \in \mathcal{R}; \quad t = 0, 1, \dots, T - 1, \tag{6}$$

where  $T = \sum_{i \in V} \max(p_i, \max_{(i,j) \in E} \delta_{ij})$  is an upper bound on the minimum project duration.

It is well known that the set  $\mathcal{S}_T$  of time-feasible schedules (which satisfy Eq. (5)) is nonempty exactly if the network does not contain a cycle of positive length (see [12]). The set  $\mathcal{S}$  of feasible schedules (which satisfy Eqs. (5) and (6)) is generally disconnected and represents the union of convex polyhedra whose number grows exponentially in  $n$ . Moreover, the decision problem whether or not  $\mathcal{S} \neq \emptyset$  is strongly NP-complete (cf. [12,144]).

Sometimes the constraints  $S_0 = 0$  and  $S_j \in \mathbb{Z}$  ( $j \in V$ ) are added to Eq. (5). We deleted these

constraints for the following reasons: For each optimal schedule  $S$  it holds that  $S_0 = 0$ . Moreover,  $S_0 = 0$  for all feasible schedules  $S$  constructed using any heuristic method discussed in Section 6.3. Since all parameters  $\delta_{ij} ((i, j) \in E)$  are integers, there always exists an integer optimal schedule provided that  $\mathcal{S} \neq \emptyset$ . All methods discussed in Sections 6.2 and 6.3 construct integer schedules.

For approximately solving  $PS | temp | C_{max}$ , a decomposition approach often turns out to be expedient. A *cycle structure* of the project network is a strong component which contains at least two nodes. For each cycle structure treated as a separate subproject with original resource capacities and started at time zero, a scheduling problem corresponding to problems (4)–(6) can be stated whose (feasible) solutions are called (feasible) *subschedules*. Neumann and Zhan [140] have proven the following theorem.

**Decomposition Theorem.** *There is a feasible schedule for the project network if and only if there is a feasible subschedule for each cycle structure.*

Several heuristic procedures for (approximately) solving project scheduling problems require a *strict order*  $\prec$  in node set  $V$  (cf. [70]). Let  $d_{ij}$  be the length of a longest path from node  $i$  to node  $j$  in the project network, where  $d_{ij} = -\infty$  if there is no path from  $i$  to  $j$ . For  $i, j \in V, i \neq j$ , we then define  $i \prec j$  if and only if either (a)  $d_{ij} > 0$  or (b)  $d_{ij} = 0$  and  $d_{ji} < 0$ .

## 6.2. Branch-and-bound methods

The basic idea of branch-and-bound algorithms for solving  $PS | temp | C_{max}$  is as follows. An optimal solution to the resource relaxation of  $PS | temp | C_{max}$  (i.e. problems (4) and (5)), for example, the *earliest schedule*  $ES = (ES_j)_{j \in V}$  with  $ES_j = d_{0j}$ , can be found in polynomial time. Starting with schedule  $S = ES$ , resource conflicts at points in time  $t$ , that is,

$$\sum_{j \in F} r_{jk} > R_k \quad \text{for some } k \in \mathcal{R} \text{ and } F \subseteq \mathcal{A}(S, t) \quad (7)$$

can be resolved successively by introducing additional temporal constraints which delay one or several activities. Set  $F$  in Eq. (7) is called a *forbidden set*. If  $F$  is minimal with respect to set inclusion, it is termed a *minimal forbidden set*.

Bartusch et al. [12], De Reyck [51], and De Reyck and Herroelen [54] have used “ordinary” precedence constraints of the type  $S_j \geq S_i + p_i$ , which correspond to adding arcs  $(i, j)$  with weight  $p_i$  to the network. Schwindt [174] has introduced *disjunctive precedence constraints*

$$S_j \geq \min_{i \in F \setminus \{j\}} (S_i + p_i), \quad (8)$$

where  $F$  is a minimal forbidden set. Instead of delaying only one activity  $j$ , several activities can be delayed at the same time which form a so-called *minimal delaying alternative* (cf. [51]). Then (Eq. (8)) is replaced by

$$\min_{j \in M_2} S_j \geq \min_{i \in M_1} (S_i + p_i)$$

with minimal delaying alternative  $M_2$  and  $M_1 := \mathcal{A}(S, t) \setminus M_2$ .  $M_2$  is an inclusion-minimal set containing at least one element of each minimal forbidden set  $F \subseteq \mathcal{A}(S, t)$ .

To solve  $PS | temp | C_{max}$ , Schwindt [174] has considered two partial problems. The *sequencing problem* consists of finding a set  $Q$  of schedules which satisfy disjunctive precedence constraints such that  $\emptyset \neq \mathcal{S}_T \cap Q \subseteq \mathcal{S}$ . The corresponding *scheduling problem* consists of minimizing  $S_{n+1}$  subject to  $S \in \mathcal{S}_T \cap Q$ . In contrast to the case of ordinary precedence constraints, the feasible region  $\mathcal{S}_T \cap Q$  of the latter problem is no longer convex if disjunctive precedence constraints are used, but represents the union of convex polyhedra. A pseudopolynomial fixed-point algorithm for solving the scheduling problem has been devised by Schwindt [174].

The branch-and-bound algorithm then consists of appropriately enumerating sequencing solutions  $Q_1, \dots, Q_r$  with  $\bigcup_{v=1}^r (\mathcal{S}_T \cap Q_v) = \mathcal{S}$  such that

$$\min_{S \in \mathcal{S}} S_{n+1} = \min_{v=1, \dots, r} \min_{S \in \mathcal{S}_T \cap Q_v} S_{n+1}.$$

Preprocessing procedures as well as good lower bounds and fathoming rules speed up the branch-and-bound method (see [174]). An overview of recent



preprocessing techniques as well as constructive and destructive lower bounds for  $PS \mid temp \mid C_{\max}$  can be found in Heilmann and Schwindt [89].

The concept of disjunctive precedence constraints markedly reduces the number of enumeration nodes of the search tree to be investigated in comparison with the branch-and-bound method by De Reyck [51]. An *experimental performance analysis* by Schwindt [174] based upon 1080 problem instances with 100 activities and five resources each (generated by the problem generator ProGen/max by Schwindt [174]) has shown that Schwindt's method solves more instances to optimality within 10 s than De Reyck's method in 100 s (using an IBM-compatible PC Pentium 200).

Another branch-and-bound procedure for  $PS \mid temp \mid C_{\max}$  has been investigated by Möhring et al. [135]. The main difference lies in the way of resolving resource conflicts. Contrary to the procedures proposed by Bartusch et al. [12], De Reyck [51], De Reyck and Herroelen [54] and Schwindt [174], where additional (disjunctive) precedence constraints are introduced to resolve a conflict, the idea is to introduce ordinary *release dates* instead. That is, a resource conflict at a certain time  $t$  is resolved by increasing the release dates  $d_{0j}$  of activities  $j \in M_2$  (i.e. the time lags between activities 0 and  $j \in M_2$ ) according to

$$d_{0j}^{\text{new}} := \min_{i \in M_1} (S_i + p_i) \quad \text{for all } j \in M_2,$$

where  $M_2$  is a minimal delaying alternative and  $M_1 := \mathcal{A}(S, t) \setminus M_2$ . Every node of the enumeration tree is then represented only by a vector of release dates (or start times, respectively), and, except for the values  $d_{0j}$ , the path lengths  $d_{ij}$  ( $i, j \in V$ ) remain unchanged in the course of the algorithm.

On the one hand, this may in principle enlarge the enumeration tree considerably as has also been observed by Schwindt [174]. Since no precedence relation is introduced, neither “ordinary” nor disjunctive, it may happen that the same resource conflict has to be resolved several times, due to the existence of maximal time lags. But on the other hand, this way of branching gives rise to a remarkable speedup in the computation of time-feasible schedules once a branching has been performed. More precisely, the computation of opti-

mal time-feasible schedules, and the corresponding lower bounds for newly generated nodes is then linear in the number of activities for every release date that has been increased. This is a major advantage over the procedures that introduce precedence constraints, where the complexity for the computation of time-feasible schedules is quadratic in the number of activities for every added precedence constraint (see e.g. Bartusch et al. [12]), and pseudopolynomial in the case of disjunctive precedence constraints (see the above-mentioned *scheduling problem* and [174]).

The disadvantage of multiple occurrence of the same resource conflicts, and the corresponding growth of the enumeration tree is tried to be kept small by performing *immediate selection* rules as well as a (surprisingly simple) dominance rule. Computational results indicate that, even without implementation of more sophisticated lower bounds, the procedure is competitive with the ones proposed by Schwindt [174] and De Reyck and Herroelen [54].

Recently, Dorndorf et al. [57] used constraint propagation techniques for  $PS \mid temp \mid C_{\max}$ . They showed that an integration of further constraints on the start times in the aforementioned sense within a new time-oriented branching scheme provides very promising results.

### 6.3. Heuristic procedures

To solve large instances of  $PS \mid temp \mid C_{\max}$  approximately, truncated branch-and-bound techniques based upon Schwindt's algorithm and priority-rule methods have been developed. As to *truncated branch-and-bound procedures*, a filtered beam search technique, an  $\varepsilon$ -approximate algorithm, and a decomposition method have been proposed by Schwindt [174]. The decomposition method exploits the Decomposition Theorem from Section 6.1. First, for each cycle structure  $\mathcal{C}$  of the network, an optimal schedule  $S^{\mathcal{C}}$  is computed by the branch-and-bound algorithm. Second, each cycle structure  $\mathcal{C}$  is replaced by an equivalent cycle of length zero whose arc weights are determined using  $S^{\mathcal{C}}$ . Third, the  $\varepsilon$ -approximate algorithm is applied to the resulting network.

*Priority-rule methods* for  $PS | temp | C_{max}$  have been devised and tested by Zhan [201], Neumann and Zhan [140], Brinkmann and Neumann [26], and Franck and Neumann [70] (the last reference contains the most recent results). Two different approaches have turned out to be expedient. The *sequential* or *direct method* schedules the activities one after another without considering the cycle structures of the network separately. The *contraction method* again exploits the Decomposition Theorem. First, a feasible subschedule is determined for each cycle structure. Second, each cycle structure is replaced by a single node or activity, respectively, with appropriate duration and (time-dependent) resource usage. Third, a feasible schedule for the resulting “contracted” network without cycles is computed. Fourth, a feasible schedule for the original network is determined using the schedules for the contracted network and the individual cycle structures.

To find a feasible schedule (for the whole network, the contracted network, or a cycle structure), a *serial* and a *parallel schedule generation scheme* have been developed. Among a large number of priority rules, the LST rule has turned out to be best. That is, the activity to be scheduled next is always an “eligible” activity (all of its predecessors with respect to strict order  $\prec$  have already been scheduled) with smallest latest start time (cf. [70]). To take maximum time lags into account, both generation schemes contain a *backward scheduling process* which is as follows: If the earliest resource-feasible start time of the activity  $j$  to be scheduled exceeds the latest possible start time of  $j$  induced by some maximum time lag  $d_{ij}^{max}$ , the start time of activity  $i$  (and of some additional activities already scheduled) has to be enlarged appropriately.

An *experimental performance analysis* based upon 120 instances with 500 activities and five resources each (generated by ProGen/max) has provided the following main results (cf. [142,174]): The priority-rule methods are much faster than the truncated branch-and-bound procedures. Whereas the direct and contraction methods require 1 and 2 s, respectively, of computing time per instance on the average (using a PC Pentium 200), the decomposition method as slowest heuristic requires almost 1 m. The decomposition and contraction

methods, which exploit the Decomposition Theorem, provide feasible (optimal) schedules for 100% (6%) and 98% (4%), respectively, of all solvable instances, where the average relative deviation of the project duration computed from the best lower bound is around 5%. The remaining heuristics solve much less instances to feasibility (the direct method only 53%), but more instances to optimality (the filtered beam search technique 62%).

## 7. Nonregular objective functions

The objective function of problem  $PS | temp | C_{max}$  discussed in Section 6 is regular, i.e. nondecreasing in the completion times of activities (in the case of a minimization problem). In this section, we deal with two kinds of nonregular objective functions where we again assume that general minimum and maximum start–start time lags are given. If the objective function to be minimized represents some measure of the variation of resource utilization, we speak of a *resource leveling problem*. In the *net present value problem*, the objective function represents the net present value of the project which is to be maximized.

### 7.1. Model

In addition to the temporal constraints (5) of  $PS | temp | C_{max}$ , we explicitly require that  $S_0 = 0$ ,  $S_j \in \mathbb{Z}$  ( $j \in V$ ), and there is a prescribed maximum project duration  $\bar{d} \in \mathbb{Z}_{\geq 0}$  with  $\bar{d} \geq d_{0,n+1}$ .

In the objective function of the *resource leveling problem*  $PS | temp | \sum c_k f(r_k(S, t))$ ,  $c_k > 0$  is the cost per unit of resource  $k$ . This problem can then be stated as follows:

$$\min \sum_{k \in \mathcal{R}} c_k f(r_k(S, t)) \quad (9)$$

$$\text{s.t. } S_j - S_i \geq \delta_{ij} \quad (i, j) \in E,$$

$$S_0 = 0,$$

$$S_{n+1} \leq \bar{d}, \quad (10)$$

$$S_j \in \mathbb{Z}_{\geq 0} \quad j \in V,$$

$$r_k(S, t) \leq R_k, \quad k \in \mathcal{R}; \quad t = 0, 1, \dots, \bar{d} - 1. \quad (11)$$

In Neumann and Zimmermann [141,142], three types of objective functions (9) are considered. If

$$f(r_k(S, t)) = \max_{t=0,1,\dots,\bar{d}-1} r_k(S, t), \quad (12)$$

we speak of the *resource investment problem* denoted by  $PS | temp | \sum c_k \max r_k(S, t)$ , which is used in practice when expensive resources have to be purchased. A second type of objective function where

$$f(r_k(S, t)) = \sum_{t=0}^{\bar{d}-1} [r_k(S, t) - Y_k]^+ \quad (13)$$

measures the deviation of the consumption of resource  $k$  from a target value for resource usage  $Y_k \geq 0$ .  $Y_k$  may be equal to the average resource utilization  $\sum_{j \in V} r_{jk} p_j / \bar{d}$ .  $[\dots]^+$  in Eq. (13) can be replaced by  $|\dots|$  or  $[\dots]^2$ . A third type of objective function where

$$f(r_k(S, t)) = \sum_{t=0}^{\bar{d}} [r_k(S, t) - r_k(S, t-1)]^+ \quad (14)$$

with  $r_k(S, -1) = r_k(S, \bar{d}) = 0$  considers the variation of resource utilization over time and is used, for example, if the resources represent different kinds of manpower. Again  $[\dots]^+$  in Eq. (14) can be replaced by  $|\dots|$  or  $[\dots]^2$ .

The basic concepts of cash flows and net present values of the cash flows of a project can be found in Russell [166] and Herroelen et al. [95]. In the objective function of the *net present value problem*  $PS | temp | \sum c_j^F \beta^{C_j}$ ,  $\beta$  is the discount rate per period and  $c_j^F$  the cash flow associated with activity  $j$ , which is assumed to occur at the completion time  $C_j = S_j + p_j$  of activity  $j$  and can be positive (payment received) or negative (cost incurred). The net present value problem can then be formulated as follows:

$$\begin{aligned} \max \quad & \sum_{j \in V} c_j^F \beta^{C_j} \\ \text{s.t.} \quad & (10), (11) \end{aligned}$$

### 7.2. Exact algorithms

As with problem  $PS | temp | C_{\max}$ , testing whether there is a feasible solution to  $PS | temp |$

$\sum c_k f(r_k(S, t))$  or  $PS | temp | \sum c_j^F \beta^{C_j}$  is strongly NP-complete.

For the *net present value problem*  $PS, \infty | temp | \sum c_j^F \beta^{C_j}$  (that is, there are no resource constraints (11)), De Reyck [51] has proposed a recursive search procedure which runs in  $O(n^4)$  time. This algorithm generalizes methods for problem  $PS, \infty | prec | \sum c_j^F \beta^{C_j}$  (that is, there are only minimum time lags  $d_{ij}^{\min} = p_i$ ) devised by Grinold [78], Elmaghraby and Herroelen [68], and Herroelen et al. [93]. De Reyck's recursive procedure starts with the earliest schedule  $ES$  and tries to delay firstly activities  $j$  with  $c_j^F < 0$  and secondly sets of connected activities with negative net present value as far as possible without violating the temporal constraints (10) in order to increase the net present value of the project. An *experimental performance analysis* has shown that, on the average, an instance with 100 activities can be solved in less than 1 s using a PC Pentium 60.

For the general net present value problem  $PS | temp | \sum c_j^F \beta^{C_j}$ , De Reyck [51] has devised a branch-and-bound method, which is based upon De Reyck's branch-and-bound algorithm for  $PS | temp | C_{\max}$ , where the resource-unconstrained scheduling problems with net present value as objective function are solved by De Reyck's recursive procedure for  $PS, \infty | temp | \sum c_j^F \beta^{C_j}$ . An *experimental performance analysis* with instances with up to 50 activities and five resources has shown that, in principle, the branch-and-bound method for  $PS | temp | \sum c_j^F \beta^{C_j}$  has the same effectiveness and efficiency as the corresponding procedure for  $PS | temp | C_{\max}$ . For  $PS | prec | \sum c_j^F \beta^{C_j}$ , Icmeli and Erengüç [98] have proposed a similar branch-and-bound algorithm, which introduces additional precedence constraints to resolve resource conflicts in analogy to the branch-and-bound procedure by Demeulemeester and Herroelen [48] for  $PS | prec | C_{\max}$ .

For the *resource leveling problem*  $PS, \infty | prec | \sum c_k f(r_k(S, t))$  with special objective functions, exact algorithms based upon enumeration, integer programming, or dynamic programming have been proposed by Ahuja [2], Easa [62], Bandelloni et al. [7], and Younis and Saad [200]. For  $PS, \infty | temp | \sum c_k f(r_k(S, t))$ , a time-window based branch-and-bound procedure has been

devised by Zimmermann and Engelhardt [203]. This algorithm exploits the fact that given a partial schedule  $S' = (S_i)_{i \in V'}$  with  $V' \subset V$ , for an unscheduled activity  $j \in V \setminus V'$ , there is a *time window*

$$\mathcal{T}_j = \{ES_j^{S'}, ES_j^{S'} + 1, \dots, LS_j^{S'}\}, \quad (15)$$

where

$$\begin{aligned} ES_j^{S'} &= \max(d_{0j}, \max_{i \in V'}(S_i + d_{ij})), \\ LS_j^{S'} &= \min(\bar{d} - d_{j,n+1}, \min_{i \in V'}(S_i - d_{ji})). \end{aligned} \quad (16)$$

The nodes of the enumeration tree correspond to partial schedules  $S'$  with the root corresponding to  $S_0 = 0$ . At a node representing partial schedule  $S'$ , the algorithm branches as follows: Select an activity  $j \in V \setminus V'$  with minimum  $LS_j^{S'} - ES_j^{S'}$  and, for each  $t \in \{ES_j^{S'}, \dots, LS_j^{S'}\}$ , generate a child  $S''$  by setting  $S_j'' = t$ . For so-called  $r$ -monotonous objective functions (which include functions of types (12) and (13) with  $Y_k = 0$ ), good lower bounds at the nodes  $S'$  can be computed. For a generalization of this branch-and-bound algorithm to problem  $PS | temp | \sum c_k f(r_k(S, t))$  with resource constraints (11) and a preliminary performance analysis we refer to Zimmermann and Engelhardt [203].

For the *resource investment problem*  $PS | temp | \sum c_k \max r_k(S, t)$ , Nübel [143] has proposed a branch-and-bound procedure in analogy to the algorithm by Schwindt [174] for  $PS | temp | C_{\max}$ . For the nodes of the enumeration tree, which correspond to sequencing solutions, fictitious maximum resource capacities are introduced to decrease the resource capacity levels required. Resulting fictitious resource conflicts are again resolved by adding disjunctive precedence constraints. The special case  $PS | prec | \sum c_k \max r_k(S, t)$  is the subject of De-meulemeester [47] and Möhring [127].

### 7.3. Heuristic procedures

For the *net present value problem*  $PS | prec | c_j^F \beta^{C_j}$ , several priority-rule heuristics have been proposed, for example, by Russell [167]

and Padman and Smith-Daniels [152]. These papers also contain an experimental performance analysis for problem instances with 1000 or more activities and several resources. A simulated annealing approach has been presented and compared with priority-rule methods using stochastic scheduling rules by Yang et al. [199]. For the *resource leveling problem*  $PS, \infty | prec | c_k f(r_k(S, t))$  with special objective functions, pseudopolynomial priority-rule methods have been devised by Burgess and Killebrew [33], Harris [82,83], Takamoto et al. [191], and Savin et al. [170]. Only small problem instances with up to 20 activities have been solved (approximately) by those methods. For problem  $PS, \infty | temp | \sum c_k f(r_k(S, t))$ , pseudopolynomial heuristics have been proposed and tested for instances with up to 100 activities and several resources by Brinkmann and Neumann [26].

Several variants of a *polynomial priority-rule method* have recently been presented by Zimmermann [202] and Neumann and Zimmermann [141,142], which can be applied to both  $PS | temp | \sum c_k f(r_k(S, t))$  and  $PS | temp | c_j^F \beta^{C_j}$ . We briefly sketch the basic idea of that procedure. At first we consider the case without resource constraints (11). Given a partial schedule  $S' = (S_i)_{i \in V'}, V' \subset V$ , the activity to be scheduled next is either a critical activity  $j \in V \setminus V'$  (that is, with slack time equal to zero) or, if there is none, an activity  $j \in V \setminus V'$  with highest priority. For the net present value problem, the priority rule greatest absolute value of cash flow (GCF) is recommended. The start time  $S_j$  of activity  $j$  to be scheduled next equals  $ES_j^{S'}$  for  $c_j^F \geq 0$  and  $LS_j^{S'}$  for  $c_j^F < 0$  (where  $ES_j^{S'}$  and  $LS_j^{S'}$  are again given by Eq. (16)). For the resource leveling problem, the priority rules MSO (minimum number of predecessors with respect to strict order  $\prec$ ), GRD (greatest resource demand  $p_j \sum_{k \in \mathcal{A}} r_{jk}$ ), MST (minimum slack time), and LST (smallest latest start time) are appropriate, where the “best” rule depends on the type of objective function (cf. [141,142]). The start time  $S_j$  of activity  $j$  to be scheduled next is a minimizer of a penalty function which represents the additional cost arising when activity  $j$  is scheduled at time  $S_j$  on a certain decision set  $\mathcal{D}_j$ .  $\mathcal{D}_j$  is a subset of the time window  $\mathcal{T}_j$

(see (15)) whose cardinality is linear in  $n$  and which depends on the objective function.

Two methods of generalizing the above priority-rule procedure to the case where there are resource constraints (11) are described in Neumann and Zimmermann [141,142]. An *experimental performance analysis* has shown that, on the average, an instance of problem  $PS, \infty \mid temp \mid \sum c_k f(r_k(S, t))$  with 500 activities and five resources can (approximately) be solved in less than 2 s using a PC Pentium 200. For an instance of  $PS \mid temp \mid \sum c_k f(r_k(S, t))$  with 200 activities and five resources, the average running time is less than 1 s. The running times for instances of the net present value problem are much smaller.

## 8. Stochastic activity durations

In real life projects, it usually does not suffice to find good schedules for fixed deterministic processing times, since these times mostly are only rough estimates and subject to unpredictable changes due to unforeseen events (weather conditions, obstruction of resource usage, delay of predecessors of an activity etc.).

In order to cope with such influences, the processing time of an activity  $j$  is assumed to be a random variable  $p_j$ . Then  $\mathbf{p} = (p_1, p_2, \dots, p_n)$  denotes the (random) vector of processing times, which is distributed according to a joint probability distribution  $P$ . In principle, this distribution  $P$  is assumed to be known (though, as will become clear later, there are methods that can deal with incomplete information about the distribution). Moreover, there may be stochastic dependencies between the different individual processing times  $p_j$ , which are represented by the joint distribution  $P$ . In our classification, these problems are denoted by  $PS \mid prec, p_j = sto \mid C_{\max}$ .

The necessity of involving stochastic methods into project planning becomes obvious if one compares the “deterministic makespan”  $C_{\max}(E(p_1), \dots, E(p_n))$  obtained from the expected processing times  $E(p_j)$  with the expected makespan  $E(C_{\max}(\mathbf{p}))$ , even in the absence of resource constraints. There is a systematic underestimation

$$C_{\max}(E(p_1), \dots, E(p_n)) \leq E(C_{\max}(\mathbf{p}_1, \dots, \mathbf{p}_n)),$$

which may become arbitrarily large with increasing number  $n$  of activities or, for fixed  $n$ , increasing variances of the processing times (see [90]). Equality holds if and only if there is one path that is critical with probability 1. This systematic underestimation of the expected makespan has already been observed by Fulkerson [75]. The error becomes even worse if one compares the deterministic value  $C_{\max}(E(p_1), \dots, E(p_n))$  with quantiles  $t_q$  such that  $Prob\{C_{\max}(\mathbf{p}) \leq t_q\} \geq q$  for large values of  $q$  (say  $q = 0.9$  or  $0.95$ ). This is the reason why good practical planning tools should incorporate stochastic methods. An overview about these methods is given in Section 8.1. Section 8.2 then deals with random processing times in the presence of resource constraints.

### 8.1. Stochastic scheduling without resource constraints

Due to the practical importance of stochastic scheduling, many methods have been developed over the last 35 years. For stochastically independent processing times, these methods can be roughly grouped into *simulation* methods (e.g. [34,176,180,190]), methods for *bounding or calculating the expected makespan* (e.g. [55,59,63,65,76,163]), methods for *analyzing the “most critical” path* (e.g. [181,182]), and methods for *bounding the whole distribution function of the makespan* (e.g. [56,81,107,175,183]).

An overview of the knowledge and the mathematical tools up to 1989 has been given by Möhring and Radermacher [131].

Most of these contributions have not been aware of the enormous inherent complexity of the problem, which was formally analyzed only 1988 by Hagstrom [80]. She considers the following two problems:

*MEAN*: Given a project network with discrete, independent processing times  $p_j$ , compute the expected makespan  $E(C_{\max}(\mathbf{p}))$ .

*DF*: Given a project network with discrete, independent processing times  $p_j$  and a time  $t$ ,

compute the probability  $Prob\{C_{\max}(\mathbf{p}) \leq t\}$  that the project finishes by time  $t$ .

Hagstrom shows that the 2-state versions of these problems, in which every processing time  $p_j$  has only two discrete values, are  $\#\mathcal{P}$ -complete (any  $\#\mathcal{P}$ -complete problem is polynomially equivalent to counting the number of Hamiltonian cycles of a graph and thus in particular NP-complete). This result is derived from a fundamental result of Provan and Ball [161] on the  $\#\mathcal{P}$ -completeness of reliability problems and shows another connection of project scheduling to reliability theory besides time–cost tradeoff problems.

The complexity status of the general version of MEAN is open (only the 2-state version, which has a short encoding, is  $\#\mathcal{P}$ -complete). If the processing times  $p_j$  may take more than 2 values, the problem has a longer encoding that in principle could admit a polynomial algorithm for solving MEAN. This is, however, not the case for DF. But also for MEAN, Hagstrom provides some evidence that problems with a long encoding may still be difficult, since MEAN and DF cannot be solved in time polynomial in the number of values of  $C_{\max}(\mathbf{p})$  unless  $\mathcal{P} = \mathcal{NP}$ .

These results show that efficient methods for calculating the expected makespan or quantiles of the distribution function of the makespan are very unlikely to exist, and thus (although in retrospect) justify the great interest in approximate methods such as bounds, simulation etc.

Many of the methods that provide bounds for the distribution function of the makespan transform the given network (mostly represented as an activity-on-arc network) into a series–parallel network that is more easily evaluated since series and parallel reductions of two activities  $h, j$  in the network correspond to the convolution  $F_h * F_j$  and pointwise product  $F_h \cdot F_j$  of their processing time distribution functions  $F_h, F_j$ , respectively. Typical examples in this respect are the bounds by Dodin [56], Kleindorfer [107] and Spelde [183].

Möhring and Müller [128] give a unified model for such bounding results in terms of a *chain-minor* notion for project networks. A network  $G_1 = (V_1, E_1)$  is a *chain-minor* of a network  $G_2 = (V_2, E_2)$  if (1) and (2) below hold.

- (1) Every activity  $j \in V_1$  is represented in  $G_2$  by a set of *copies* or *duplicates*  $D(h)$ , where  $D(h) \cap D(j) = \emptyset$  if  $h \neq j$ .
- (2) Every chain  $C$  (the set of activities on a path) of  $G_1$  is “contained” in a chain  $C'$  of  $G_2$  in the sense that, for every activity  $j \in C$ , there is a duplicate  $j' \in D(j)$  with  $j' \in C'$ . (These duplicates  $j$  may be different for different chains  $C$  of  $G_1$ .)

Möhring and Müller [128] show that, if  $G_1$  is a chain-minor of  $G_2$ , then one obtains a lower bound for the distribution function  $F_{G_1}$  of the makespan of  $G_1$  if one gives every duplicate  $j'$  of an activity  $j$  the same processing time distribution as activity  $j$ , treats them as independent, and calculates the distribution function  $F_{G_2}$  of the makespan of  $G_2$ . In other words,

$$Prob\{C_{\max} \leq t \text{ in } G_1\} \geq Prob\{C_{\max} \leq t \text{ in } G_2\}$$

for every  $t$ .

This very general bounding principle covers the mentioned specific bounds of Kleindorfer, Spelde, Dodin and others. Moreover, if one can identify networks  $G_1, G_2$  that “sandwich” the given network  $G$  in the sense that  $G_1$  is a chain-minor of  $G$  and  $G$  is a chain-minor of  $G_2$ , then the unknown makespan distribution function  $F_G$  of  $G$  is “sandwiched” by those of  $G_1$  and  $G_2$ , i.e.,  $F_{G_1} \geq F_G \geq F_{G_2}$ .

This brings up the question to identify networks  $G_1$  and  $G_2$ , for which the distribution functions  $F_{G_1}$  and  $F_{G_2}$  are easier to evaluate. If  $G_1$  and  $G_2$  are chosen to be series–parallel, then the computation reduces to a sequence of convolutions and products of distribution functions. Möhring and Müller [128] show that the 2-state version of DF is still  $\mathcal{NP}$ -complete, but only in the weak sense. However, MEAN can in this case be solved in time polynomial in the largest number of values of the makespan of a network encountered in any series–parallel reduction sequence.

The quality of these bounds depends on the “distance” of the given network  $G$  from being series–parallel, and is another motivation for studying distance measures as the reduction complexity and the factoring complexity discussed in connection with time–cost tradeoff problems in Section 4.2. In fact, any activity duplication in the

sense discussed there, leads to a lower bound for the makespan distribution function by the chain-minor result. This is the driving principle behind the bound of Dodin [56].

Another way to facilitate the calculation of the makespan distribution function is, similar to time–cost tradeoff problems, the substitution or modular decomposition. The first rigorous analysis of modular decomposition in connection with stochastic networks was done by Radermacher [162], see also [131]. It can be used both for exact calculation and for bounds.

All the methods discussed above assume that the processing time distributions are known, which usually is not the case in practice and often inhibits the use of these methods. A way to cope with this *incomplete information* is offered by the bounds of Spelde [183]. He takes as network  $G_2$  in the “sandwich” a series composition of all paths of  $G$ , thus duplicating an activity as many times as it is contained in a path of  $G$ . The distribution function  $F_{G_2}$  of the resulting series–parallel network  $G_2$  is then the product of the distribution functions of the lengths of these paths, say  $F_{G_2} = F_1 \cdot F_2 \cdot \dots \cdot F_N$ , where  $F_i$  is the distribution function of the  $i$ th path.

If the network  $G$  is large enough, i.e. all paths contain “enough” activities, then, by the central limit theorem, every  $F_i$  is approximately a normal distribution function, whose mean  $\mu_i$  and variance  $\sigma_i^2$  are obtained as the sum of the means and variances of the processing times  $p_j$  of all activities  $j$  contained in the  $i$ th path.

Hence it suffices to know the expected processing time  $E(p_j)$  and the variance  $V(p_j)$  of every activity in order to calculate the Spelde bound. There is, however, a complication since the number  $N$  of all paths may be exponential in the size of the given network  $G$ . This can be overcome by calculating the first  $k$  longest paths w.r.t. expected processing times  $E(p_j)$ , until  $\text{Prob}\{k\text{th path is longer than } l \text{ st path}\} \leq \varepsilon$  for a given accuracy parameter  $\varepsilon$  (say  $\varepsilon = 0.05$ ). If  $F_1, F_2, \dots, F_k$  are the normal distribution functions of these paths, then  $F_{G_2} \approx F_1 \cdot F_2 \cdot \dots \cdot F_k$ . In practice  $k$  will be small.

In fact, this method contains the traditional PERT as a special case, since PERT only analyzes the distribution of the path with the longest expected path length.

Ludwig et al. [121] have implemented several of these bounds (Kleindorfer, Dodin, Spelde) and have made an extensive computational study of their bounding behavior on networks having up to 500 activities. The main conclusions from this study are that the Spelde bounds provide an excellent approximation that can be computed very fast. It usually overestimates the quantiles  $t_q$  for  $q \in [0.9, 1]$  only by about 5%, and thus provides a very good practical planning tool. The bounds of Dodin and Kleindorfer have an even smaller overestimation, but require complete knowledge of the processing time distributions. The accuracy of the bounds can be improved to less than 2% overestimation through the use of decomposition.

For dependent processing times, the above bounds cannot be used. Instead, there is a different approach that calculates an upper bound for the expected tardiness  $E[\max\{0, C_{\max}(\mathbf{p}) - t\}]$  as function of  $t$ , which is valid for any joint distribution of the processing times and hence for all possible dependencies among them.

This approach has been investigated by Klein Haneveld [106], Meilijson and Nadas [124] and Weiss [198]. Interestingly, for discrete processing time distributions, the evaluation of this bound can be interpreted as a time–cost tradeoff problem with piecewise linear and convex cost functions that is equivalent to a linear time–cost tradeoff problem as discussed in Section 4.2. An overview of this bound is given in [131].

Another, more recent extension of stochastic network analysis concerns the combination of random processing times with the time–cost tradeoff paradigm. Here, one influences the processing time distribution of an activity by allocating more resources (money) to it. One then wants to minimize the expected makespan (or other distribution parameters) subject to a fixed budget. We refer to the work of Bowman [25] and Foldes and Soumis [69] for details.

## 8.2. Stochastic scheduling with resource constraints

We now consider random processing times together with resource constraints as for  $PS \mid prec \mid C_{\max}$ . This combination has often been

studied in machine scheduling, but much less in project scheduling. The model leads into the area of stochastic dynamic programming. Scheduling is done by *policies* or *strategies*. A complete characterization of all policies and subclasses thereof has been given by Möhring et al. [132,133].

For stability reasons explained there, only so-called *elementary strategies* are applicable in practice. Such a policy  $\Pi$  has the following dynamic interpretation. It chooses actions at decision points. *Decision points* are  $t = 0$  (project start) and activity completions. An action at time  $t$  consists of starting a *feasible set*  $S(t)$  at  $t$ , where feasible means that precedence and resource constraints are respected. The decision may of course only exploit information that has become available until the current time  $t$ .

In the end, when every activity has been scheduled, we have a realization  $p$  of processing times and  $\Pi$  has constructed a schedule  $\Pi(p) = (S_1, S_2, \dots, S_n)$  of starting times for the activities.  $C_{\max}^{\Pi}(p)$  denotes the makespan of that schedule, and  $E(C_{\max}^{\Pi}(p))$  the expected makespan under policy  $\Pi$ . The aim then is to find a policy that minimizes the expected makespan.

Policies may be classified according to how they solve the resource conflicts. This can be modeled by looking at the set  $\mathcal{F}$  of *minimal forbidden sets*  $F \subseteq V$ . Every proper subset  $F' \subset F$  of such a set  $F \in \mathcal{F}$  can in principle be scheduled simultaneously, but the set  $F$  itself cannot because of the resource constraints.

A natural class of policies is the class of *preselective policies* introduced by Igelmund and Rademacher [99,100]. They solve the resource conflict on every forbidden set  $F \in \mathcal{F}$  by choosing an activity  $j_F \in F$  that can only start after some other activity  $j \in F \setminus \{j_F\}$  has finished. This idea has also recently been used in deterministic scheduling under the name of *delaying alternatives*, see also Section 6.2.

A subclass of the class of preselective policies is obtained by letting the selected activity  $j_F \in F$  always wait for the same activity  $i_F \in F \setminus \{j_F\}$ . Any such policy  $\Pi$  can be identified with a network  $G'$  constructed from the given network  $G$  by adding all such precedence constraints  $i_F \rightarrow j_F$ . The policy  $\Pi$  then constructs as  $\Pi(p)$  the earliest start

schedule of  $G'$  for processing times  $p$ . These policies are therefore called ES-policies (Earliest Start policies).

Stork [189] has implemented a branch-and-bound algorithm for both classes of policies that finds an optimal preselective policy or ES-policies for stochastified ProGen instances with up to 20 activities in reasonable time. Unlike branch-and-bound algorithms for the associated deterministic setting, these algorithms require knowledge of the set  $\mathcal{F}$  of minimal forbidden sets in advance and cannot use lower bounding techniques or dominance rules that involve knowledge of all processing times.

Motivated by the precedence tree concept used in branch-and-bound algorithms for  $PS | prec | C_{\max}$ , see Section 3.1., Möhring and Stork [134] identify an interesting subclass of the class of preselective policies, the *linear preselective policies*. Such a policy  $\Pi$  chooses the waiting activities  $j_F \in F$  as the last activity in  $F$  according to a predefined linear ordering on the set  $V$  of activities that is a topological sorting of the graph  $G$  of precedence constraints.

This class of policies leads to a significant speedup in computation time, since the calculation of the expected completion time can be done more efficiently, and since many preselective policies that are dominated by others are no longer generated. Moreover, it is possible to efficiently decide whether the conflict on a currently considered forbidden set has already been indirectly settled by previous choices of waiting activities  $j_F$  for other forbidden sets. These properties make it currently possible to solve most of the ProGen instances with up to 30 activities to optimality.

## 9. Further models

Enterprises are and have been facing mounting pressures to exercise reductions in costs arising from producing their goods or services and to make better use of existing staff or equipment. It is well known from practical experience that this pressure can be met, at least in part, by more efficient and intelligent planning. Successful application of these methods, however, depends to a



large degree on the ability to unambiguously and efficiently model the relevant specifics of the problems tackled. This ability, in turn, calls for expressive modeling concepts, which allow to capture a wide range of requirements appearing in real world problems. Additionally, advanced methods exploiting the degree of freedom covered by advanced models are also necessary.

Obviously, the models and methods discussed so far in this paper meet these requirements to some extent. However, there are numerous practical problem settings which require more general models. Some of them are mentioned in what follows.

(i) Recently, it has been shown by Dayanand and Padman [43,44] that the usual approach to relate to one single model which has to cover both the contractor's and the client's view of the problem might not be appropriate in practice. Consequently, in [43] models for the contractor and in [44] models for the client are discussed.

(ii) A generalization of  $PS | prec | C_{\max}$  is considered in Böttcher et al. [21]. There so-called partially renewable resources are defined by assuming for each resource a capacity on subsets of periods. In [21] exact branch-and-bound and serial heuristic algorithms have been developed. The concept of partially renewable resources is a fundamental tool in order to make e.g. timetabling and shift scheduling amenable to project scheduling. In addition, partially renewable resources serve to model complicated labor regulations. Furthermore, they cover traditional renewable and nonrenewable resource constraints as special cases. Finally, in Schirmer and Drexl [171] it is shown that partially renewable resources can be used to express several kinds of logical relations between the scheduling of activities. In addition, a number of practical requirements on activities' scheduling can be formulated such as maximum or minimum quotas, as well as issues of calendarization. This underscores the expressive power of partially renewable resources.

(iii) In the multi-mode case of project scheduling all mode-activity-assignments are mutually independent in the sense that assigning a mode to one activity  $j$  of a project consisting of  $n$  nonpreemptable activities does not necessarily force any other activity to be processed in a specific mode. In some applications this is not feasible. Imagine, e.g., a

situation in which certain activities belong together in the sense that they must be executed in the same way. This leads to the mode identity case that has been recently introduced into the project scheduling literature by Salewski et al. [168]. There it is proven that  $MPS | prec | C_{\max}$  is a special case of the more general mode identity case. Moreover, it is shown that the (feasibility variant of the) mode identity case is strongly ( $\mathcal{NP}$ -complete)  $\mathcal{NP}$ -hard. Furthermore, greedy randomized adaptive search procedures are presented. Finally, it is shown that the mode identity case serves to model applications to audit-staff scheduling.

## 10. For further reading

[3,27,41,52,53,64,150,156,173,184,192,193]

## Acknowledgements

This research has been supported by Deutsche Forschungsgemeinschaft Grants Br 389/15, Dr 170/6, Mo 446/3, Ne 137/4, Pe 514/7. The authors are indebted to Alf Kimms, Frederik Stork and Marc Uetz for careful reading previous versions of the manuscript.

## Appendix A. Constraint propagation techniques

So far we have discussed methods which are tailored for solving, e.g.,  $PS | prec | C_{\max}$ . Now, a short review of the basis of constraint propagation, the sequence consistency tests which have primarily been developed for solving the special case  $PSm, 1, 1 | prec | C_{\max}$  will be given. Though designed for the special case, it is strongly conjectured that these tests are also applicable to the more general problem  $PS | prec | C_{\max}$ . We start with the description of some basic concepts.

### A.1. Basic concepts

The scope of inference or propagation techniques is to reach a certain level of consistency in

order to accelerate exact algorithms or local search procedures. Model based local reasoning over the constraint set makes problem specific knowledge, which is implicitly contained in the model description, explicitly available.

Most existing knowledge based scheduling systems are only capable of incorporating a small fraction of scheduling knowledge. Encouraged by this little success and the progress that is made in the development of general problem solvers in form of constraint based logic programming languages (cf. ILOG, see [120]), here we are going to restrict ourselves to the problem  $PSm, 1, 1 | prec | C_{max}$ . Recognizing some typical features probably could enormously increase the power of general problem solvers in order to solve optimization, and in particular constraint satisfaction problems.

A constraint satisfaction problem (CSP) consists of a set of  $n$  variables  $Y_1, \dots, Y_n$ , their domains  $D_1, \dots, D_n$ , respectively, and a set of constraints of these variables. An  $n$ -ary relation or constraint on  $Y_1, \dots, Y_n$  is a subset of the cartesian product  $D_1 \times D_2 \times \dots \times D_n$  of the domains. A solution is a value assignment of the variables such that all constraints are satisfied. As a special case a binary CSP consists only of constraints on two variables (cf. [122,125]). Obviously, project scheduling problems can be considered as constraint satisfaction problems whereby the objective function is included into the set of constraints.

A graph may serve as an illuminating representation of constraint satisfaction problems. In the dual representation each vertex of the graph corresponds to a constraint and vertices are adjacent if the vertices representing constraints have at least one variable in common. In the primal representation – the only one we are going to consider – each vertex of the graph corresponds to some variable of the CSP. An edge is a subset of the vertex set. The edge represents precisely those constraints which constitute of these and only these variables represented by the edge defining vertex set. Hence, an edge implicitly is defined by the set of all feasible tuples of variable instantiations of the edge defining constraints. The resulting graph is a hypergraph. The situation is much simpler in case of a binary CSP, i.e. a constraint

$CON_{ij}(Y_i, Y_j)$  contains at most two variables  $Y_i$  and  $Y_j$ , and corresponds to a subset of the Cartesian product  $D_i \times D_j$ . An edge connecting the vertices of the variables  $Y_i$  and  $Y_j$  corresponds to all binary relations on these two variables. The resulting graph is said to be the constraint graph of the underlying CSP. A universal relation between any two variables  $Y_i$  and  $Y_j$ , i.e. constraints which are satisfied by all tuples of the Cartesian product  $D_i$  and  $D_j$ , is not included into the graph. Universal constraints do not deliver any information.

Some simple consistency checks at the beginning of the search can drastically reduce the size of the search tree. These tests of consistency have the advantage that the constraint graph becomes more explicit, i.e. hidden constraints on variables currently not adjacent in the constraint graph get visible and new edges may be introduced into the graph. Hereby we say that a set of variables is  $\alpha$ -consistent if it is  $(\alpha - 1)$ -consistent and for any subset of  $\alpha - 1$  variables and any instantiation of these  $\alpha - 1$  variables satisfying all constraints there exists a value in the domain of the remaining variable such that all constraints on all  $\alpha$  variables are satisfied. 1-consistency means that for every variable and each of its domain variables all constraints are satisfied. A set of variables is arc consistent if it is 2-consistent (this is said in relation to the constraint graph). A pair of variables  $Y_i$  and  $Y_j$  is path-consistent if for any feasible, i.e.  $CON_{ij}(Y_i, Y_j)$  respecting, instantiation  $a_i$  and  $a_j$  of  $Y_i$  and  $Y_j$  and any sequence of edges  $(CON_{i_1 i_1}, CON_{i_1 i_2}, \dots, CON_{i_m j})$  in the constraint graph there is an instantiation  $a_{i_1} \in D_{i_1}, a_{i_2} \in D_{i_2}, \dots, a_{i_m} \in D_{i_m}$  of variables  $Y_{i_1}, Y_{i_2}, \dots, Y_{i_m}$  such that all constraints  $CON_{i_1 i_1}(a_i, a_{i_1}), CON_{i_h, i_h+1}(a_{i_h}, a_{i_h+1}), CON_{i_m j}(a_{i_m}, a_j), h = 1, \dots, m - 1$ , are satisfied. Thus path-consistency means that for any pair of variables and any explicitly feasible value pair there is also a feasible variable-value assignment on each path (edge sequence) connecting this variable pair in the constraint graph (including the universal relation). The constraint graph is arc- or path-consistent if any pair of variables is arc- or path-consistent, respectively. Arc-consistency requires an  $O(e \cdot a^2)$  effort while path-consistency can be reached with an effort of  $O(n^3 a^3)$  where  $e$  is the number of edges in the constraint graph,  $a$  is the maximum number

of elements in a domain, and  $n$  is the number of variables (cf. e.g. [91]).

As mentioned earlier consistency tests yield a more explicit constraint graph. This is quite comparable to what happens during backtrack search. The enumeration only takes those constraints into account which are explicitly contained in the constraint graph, while implicitly existing edges become just visible during the search process. Finding implicitly existing constraints means to generate new knowledge in the knowledge base, it is called constraint propagation and dates back to an early idea of Waltz [197]. Clearly, the higher the level of consistency the more constraints become visible. That means, to reach arc-consistency, path- or 3-consistency can be considered as local constraint propagation. In order to reach backtrack-poor search it is necessary to make as many constraints explicit as possible, because variable instantiations which violate implicitly existing constraints usually are detected much later during the search process. Such inconsistencies then lead to backtracking. In order to reach backtrack-free search it is indispensable to make all implicitly existing constraints explicit. Montanari [136] called that the “central problem”, obviously an  $\mathcal{NP}$ -hard one. For some special cases it is possible to require sufficient conditions in order to reach a backtrack-free or a backtrack-poor search. For instance, arc-consistency with respect to a constraint graph which is a tree guarantees backtrack-free search, cf. [72,73].

Constraint propagation, i.e. local consistency checks, can reduce the enumeration procedure before searching substantially but also during the search process it may happen that the modification of a variable domain reduces the domains of other variables, of those which are connected to the former by some constraints. That may lead to a cut of some search tree branches. Certain methods are described in the literature in order to reduce the search tree. Some of these consistency tests are described in what follows.

#### A.2. Resource-based sequence consistency tests

Consider the minimum makespan problem of project scheduling with  $m$  resources each of which

is available in precisely one unit at a time, i.e. consider  $PSm, 1, 1 | prec | C_{max}$ . The job shop scheduling problem is a special case and has received considerable attention in the literature, see the surveys by Blażewicz et al. [16,17]. There are a couple of solution approaches on job shop scheduling available which we are going to describe in the more general project scheduling setting (cf. [20,29,30,36,37,147,157,158] and the annotated bibliography by Hoogeveen et al. [97]). An illuminating description of the problem is the disjunctive graph model which has been extended by Blażewicz et al. [19].

Such a disjunctive graph is a particular form of a binary constraint graph. An activity representing vertex corresponds to a variable in the constraint satisfaction model. A variable’s domain consists of all possible starting times of the activity. Conjunctive arcs describe the precedence constraints and the orientation of the arc defines certain time dependencies. A disjunctive arc pair connecting two activities  $i, j$ , which are competing for the same resource, may be replaced by an undirected edge connecting activities  $i$  and  $j$ , or variables  $Y_i$  and  $Y_j$ , respectively.  $Y_i$  is the domain variable which shall be instantiated with start times  $S_i$ . Each edge defines the constraint  $CON_{ij}(Y_i, Y_j) = CON_{ji}(Y_j, Y_i)$  corresponding to  $Y_i + p_i \leq Y_j$  or  $Y_j + p_j \leq Y_i$ , i.e. either activity  $i$  is scheduled before activity  $j$  or the other way round.

In order to increase efficiency a domain is represented as an interval of integers of possible processing starts, hence only the interval defining endpoints are considered as domain values. Thus an inconsistency of a tuple is not realized if one of the tuple entries belongs to the interior of an interval. Such an inconsistency is only excluded from the interval if the value of the tuple responsible for the inconsistency, will become an interval endpoint at a certain time. Consider again any constraint  $CON_{ij}(Y_i, Y_j)$ . Then the left bound (with respect to arc-consistency on constraint  $CON_{ij}$ ) of the domain interval  $D_i$  is defined by the temporarily earliest possible starting (release) time of activity  $i$  such that there is a possible start time of activity  $j$  within its domain  $D_j$ . The right bound (with respect to arc-consistency on constraint  $CON_{ij}$ ) of

the domain  $D_i$  is defined as the temporarily latest possible start time of activity  $i$  such that there is a possible start time of activity  $j$  in  $D_j$  and both activities can be processed.

Assume that the order in which  $i$  and  $j$  are processed is not fixed. An arc-consistent domain of activity  $i$  means that  $D_i$  is bounded to the left by the maximum of all such earliest possible start times with respect to all activities  $j$  requiring the same resource. The right bound is given by the minimum of all such latest possible start times of activity  $i$  with respect to all activities  $j$  requiring the same resource. Arc-consistency of an interior point of the interval is not guaranteed unless it becomes an interval endpoint at some time. Consider the disjunctive graph model. Then the longest path from the initial dummy activity to activity  $i$ , which defines the release date  $ES_i$  of  $i$ , is a lower bound of the left bound of an arc-consistent interval. Correspondingly, if we subtract the tail, i.e. the length of a longest path connecting activity  $i$  (including  $p_i$ ) to the fictitious termination activity in the disjunctive graph, from the makespan (or an upper bound) then we obtain an upper bound  $LS_i$  for the right endpoint of the interval. Hence, computing heads (release times) and tails generally yields only node-consistent (1-consistent) domains of possible starting points for all activities but, if one arc is selected from each disjunctive arc pair, i.e. we have an acyclic graph defining a feasible schedule, then the computation of heads and tails provides arc-consistency. Even more, without backtracking a feasible solution can be found. As the constraint graph corresponds to an activity-on-node network, these start time windows can be derived in a straightforward way. A test of path-consistency means that for any two path-consistent activities  $i$  and  $j$ , where  $i$  may be processed before  $j$ , and any third activity  $h$  either  $h, i, j$  is a possible processing sequence or  $i, h, j$  or  $i, j, h$ . All three possibilities are checked and the intervals are modified.

Let us go into more details in order to describe a class of logical tests called sequence consistency tests which are based on resource constraints. Constraint propagation finally is a full exploration of all available resource constraints in a sense that constraints are activated to reduce variable do-

main unless domain reductions are no longer possible. Hence the effect of any propagation heavily depends on the kind of constraints. The efficiency depends on the constraint activation sequence, the variable and value selection in the backtrack search (if no further domain reduction can be achieved) and the level of consistency. These sequence consistency tests reduce activity domains by ruling out infeasible start time assignments. The benefit of the tests is that they can reduce the search space and direct an algorithm towards good solutions. From now on we are only interested in the tests themselves and will not address scheduling algorithms in which they can be embedded.

We assume that all activity domains have been made node-consistent (heads and tails have been calculated), i.e. we impose an upper bound on the makespan of at least one feasible schedule. First we derive the tests for the case of unit resource capacity as, for instances, encountered in machine scheduling. Finally, the results will be generalized for arbitrary usage of resources and arbitrary resource availability. A comprehensive presentation can be found in [58].

Recall  $p_j$ , the processing time of activity  $j$ ,  $r_{jk}$ , the per period usage of renewable resource  $k$ , and  $[ES_j, LS_j]$  ( $[EC_j, LC_j]$ ), the start (completion) time window of activity  $j$ . The domain  $D_j$  is the set of all possible start times  $S_j$  of  $j$ . It is bounded by the start time window  $[ES_j, LS_j]$  but some values in the start time window may be infeasible. We will use the following shorthand description w.r.t. any activity from a subset  $W$  of the set  $V$  of activities:

$$E(W) := \min\{ES_j \mid j \in W\},$$

$$P(W) := \sum_{j \in W} p_j$$

$$C(W) := \max\{LC_j \mid j \in W\},$$

$$LS(W) := \min\{LS_j \mid j \in W\},$$

$$EC(W) := \max\{EC_j \mid j \in W\}.$$

A sequence relation  $i \rightarrow W \setminus j$  says that activity  $i$  has to be scheduled (started and finished) before the start of any activity in set  $W \setminus j$ , i.e. there is no precedence relation between  $i$  and  $j$ .

The idea behind all sequencing tests described now is to consider subsets  $W$  of activities requiring the same resource  $k$  for being processed. Within these subsets all possible activity sequences with a particular property are examined, e.g. the property that the activity sequence does not start with an activity  $j$  from  $W$ . If all such sequences are infeasible, then we can conclude that the sequence must not have the property; therefore, we could deduce that  $j$  must be first in  $W$  since all sequences where this is not the case are infeasible. The sequencing tests are presented in order of decreasing strength. While a stronger condition allows a stronger conclusion, it is at the same time more likely to be inapplicable. Except for the last one all other tests in this section are derived for disjunctive scheduling problems where all activities are mutually exclusive in the sense that they exclusively occupy any required resources throughout their processing time. This is of course the case for instances of  $PSm, 1, 1 \mid prec \mid C_{\max}$ . However, even for instances of  $PS \mid prec \mid C_{\max}$  the tests may still be used for subsets of disjunctive activities. Disjunctive tests for activity pairs lead to good lower bounds for  $PS \mid prec \mid C_{\max}$ , cf. Klein and Scholl [105].

Carlier and Pinson [37] have derived conditions under which it can be concluded that an activity  $j$  from  $W$  must be scheduled first or last in  $W$ , see also Carlier [35]. They even used the preemptive bound for calculation of tighter values  $LS(W)$  and  $EC(W)$ . If an activity  $j$  is scheduled before or after  $W \setminus j$  we may also think of  $j$  as the input or output of  $W \setminus j$ , which stipulates the name of the following condition.

*Input/output:* Let  $j$  be an activity from a subset  $W$  of the activity set  $V$  where all activities of  $W$  require one unit of the same resource during each period of processing. If  $C(W) - E(W \setminus j) < P(W)$  then  $j$  must precede all activities in  $W \setminus j$ . If  $C(W \setminus j) - E(W) < P(W)$  then  $j$  must succeed all activities in  $W \setminus j$ .

If the input condition holds we can reduce the domain of  $j$  through an update of the latest start time  $LS_j$  to  $\min\{LS_j, LS(W \setminus j) - p_j\}$ . Symmetrically, if the output condition holds, the earliest start time  $ES_j$  can be updated to  $\max\{ES_j, EC(W \setminus j)\}$ . After reducing the domain of  $j$  it

may be possible to reduce the domains of activities in  $W \setminus j$  by applying other tests. In branch-and-bound procedures that branch over disjunctive arcs, the rules may be employed to fix disjunctions, a process often called immediate selection or edge finding, cf. Brucker et al. [29].

From the input/output condition it can be deduced that an activity  $j$  must be scheduled first or last in  $W$ . The weaker input or output condition can be used to show that a precedence relation  $i \rightarrow j$  exists between an activity pair  $i, j$  from  $W$ , cf. Blaźewicz et al. [20].

*Input or output:* Let  $i$  and  $j$  be two activities from a subset  $W$  of the activity set  $V$  where all activities of  $W$  require one unit of the same resource during each period of processing. If  $C(W \setminus j) - E(W \setminus i) < P(W)$  then  $i$  must be scheduled first or  $j$  must be scheduled last in  $W$ . If  $i$  and  $j$  are two distinct activities then  $i \rightarrow j$ .

If the input or output condition holds and  $i \neq j$  we can reduce the domains of  $i$  and  $j$  through an update of the latest start time  $LS_i$  to  $\min\{LS_i, LS_j - p_i\}$ . Symmetrically, the earliest start time  $ES_j$  can be updated to  $\max\{ES_j, EC_i\}$ . If the input or output condition holds and  $i = j$  we can reduce the domain of  $j$  by the interval  $(LS(W \setminus j) - p_j, EC(W \setminus j))$ . For the case where there are only three activities in  $W$  the input or output condition allows to draw the conclusions as the  $r$ -set condition described in Brucker et al. [29] which present an  $O(n^2)$  algorithm for checking 3-set conditions. By further relaxing the test for the input or output condition, we can still draw additional conclusions in situations where the input or output condition and the input/output condition do not hold.

*Input/output negation* (cf. [8,36,147]): Let  $j$  be an activity from a subset  $W$  of the activity set  $V$  where all activities of  $W$  require one unit of the same resource during each period of processing. If  $C(W \setminus j) - ES_j < P(W)$  then  $j$  must not precede all activities in  $W \setminus j$ . If  $LC_j - E(W \setminus j) < P(W)$  then  $j$  must not succeed all activities in  $W \setminus j$ .

The input negation extracts the interval  $[0, \min\{EC_{j'} \mid j' \in W \setminus j\}]$  from the set of possible start times of  $j$ . Symmetrically, if the set output negation holds,  $j$  must precede at least one activity

in  $W$  and it extracts the interval  $[\max\{LS_j - p_j + 1 \mid j' \in W \setminus j\}, \infty]$  from the set of possible start times of  $j$ .

We now try to reason about the amount of processing time of an activity that must fall into a given interval  $[t_1, t_2]$ . The smallest amount of time during which an activity  $j$  must be executed in the interval  $[t_1, t_2]$ , say the interval processing time, is:

$$p_j[t_1, t_2] := \max\{0, \min\{p_j, t_2 - t_1, EC_j - t_1, t_2 - LS_j\}\}.$$

The processing time for a set  $W$  that must fall into  $[t_1, t_2]$  is  $P(W, [t_1, t_2]) := \sum_{j \in W} p_j[t_1, t_2]$ .

*Interval pair ordering:* Let  $i$  and  $j$  be two activities from a subset  $W$  of the activity set  $V$  where all activities of  $W$  require one unit of the same resource during each period of processing. If  $C(\{j\}) - E(\{i\}) < P(W \setminus \{i, j\}, [E(\{i\}), C(\{j\})]) + P(\{i, j\})$  then  $j \rightarrow i$ .

*Interval consistency:* Let  $j$  be an activity from a subset  $W$  of the activity set  $V$  where all activities of  $W$  require one unit of the same resource during each period of processing. Let  $t \in [ES_j, LC_j]$ . If  $t - ES_j < P(W \setminus j, [ES_j, t]) + \min\{p_j, t - ES_j\}$  then  $j$  must be delayed, i.e. its earliest possible start can be increased by  $P(W \setminus j, [ES_j, t])$ . Symmetrically, if  $LC_j - t < P(W \setminus j, [t, LC_j]) + \min\{p_j, LC_j - t\}$  then  $j$  must be preferred, i.e. its latest possible completion can be decreased by  $P(W \setminus j, [t, LC_j])$ .

The interval consistency test can be limited to earliest start and latest completion times of activities, thus providing an efficient algorithm for this consistency test, see [58].

The aforementioned consistency tests got different names by different authors, some of them are called edge-finding. The two interval tests are also known as energetic reasoning, cf. [8]. Many of these tests generalize and extend the earlier tests on job shop scheduling, e.g. Nuijten [145] and Nuijten and Le Pape [147] update time bounds of activities using ideas presented in Carlier and Pinson [37] and incorporates the tests into a constraint satisfaction framework, cf. [146].

The results for  $PSm, 1, 1 \mid prec \mid C_{\max}$  can be generalized for resources and activities with arbitrary resource availability and usage, respectively,

cf. [9–11]. Assume for simplicity that the resource capacity is constant.

*Set input/output:* Let  $j$  be an activity from a subset  $W$  of the activity set  $V$  where all activities of  $W$  require at least one unit of the same resource  $k$  during each period of processing.

$$\text{If } [C(W) - E(W \setminus j)]R_k < \sum_{i \in W} p_i r_{ik}, \text{ then } j \rightarrow W \setminus j.$$

$$\text{If } [C(W \setminus j) - E(W)]R_k < \sum_{i \in W} p_i r_{ik}, \text{ then } W \setminus j \rightarrow j.$$

Note that the meaning of  $j \rightarrow W \setminus j$  (or  $W \setminus j \rightarrow j$ ) is, that  $j$  must start before (or end after) all activities in  $W \setminus j$ .

Hence, consistency checks, or roughly speaking propagation of constraints will make implicitly defined constraints more visible and will prune the search tree in a branch and bound algorithm. Obviously,  $n$ -consistency, where  $n$  is the number of activities, immediately implies that a feasible schedule can be generated easily, however, to achieve  $n$ -consistency is in general not practicable. Moreover, worse upper bounds on the makespan of an optimal schedule will hardly reduce variable domains, i.e. only a few arc directions are fixed during the constraint propagation process. The better the bounds the more arc directions can be fixed, see also [38,39,123].

The main interest of propagation of constraints is the flexibility that results from the fact that each constraint propagates independently from the existence or non-existence of other constraints. It appears that the propagation process can be organized to guarantee that propagation can be done via longest path computation (cf. [19,20,39]).

## References

- [1] T. Ahn, S.S. Erengüç, Resource constrained project scheduling problem with multiple crashable modes, Technical Report, College of Business Administration, University of Florida, Gainesville, USA, 1995.
- [2] H.N. Ahuja, Construction Performance Control by Networks, Wiley, New York, 1976.
- [3] C. Akkan, Two heuristics based on a graph induction method for the discrete time-cost tradeoff problem, Working Paper, Koç University, Istanbul, 1997.

- [4] R. Alvarez-Valdes, J.M. Tamarit, Heuristic algorithms for resource-constrained project scheduling: A review and an empirical analysis, in: R. Sowiński, J. Węglarz (Eds.), *Advances in Project Scheduling*, Elsevier, Amsterdam, 1989, pp. 114–134.
- [5] T. Baar, P. Brucker, S. Knust, Tabu-search algorithms for the resource-constrained project scheduling problem, Working Paper, Universität Osnabrück, 1997.
- [6] M.O. Ball, C.J. Colbourn, J.S. Provan, Network reliability, in: M.O. Ball, T.L. Magnanti, C.L. Monma, G.L. Nemhauser (Eds.), *Network Models, Handbooks in Operations Research and Management Science*, vol. 8, Elsevier, Amsterdam, 1995, pp. 673–762.
- [7] M. Bandelloni, M. Tucci, R. Rinaldi, Optimal resource leveling using non-serial dynamic programming, *European Journal of Operational Research* 78 (1994) 162–177.
- [8] P. Baptiste, C. Le Pape, A theoretical and experimental comparison of constraint propagation techniques for disjunctive scheduling, *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, Montreal, Canada, 1995.
- [9] P. Baptiste, C. Le Pape, W.P.M. Nuijten, Constrained-based optimization and approximation for job-shop scheduling, *Proceedings of the AAAI-SIGMAN Workshop on Intelligent Manufacturing Systems, 14th International Joint Conference on Artificial Intelligence (IJCAI)*, Montreal, Canada, 1995.
- [10] P. Baptiste, C. Le Pape, W.P.M. Nuijten, Incorporating efficient operations research algorithms in constrained-based scheduling, *Proceedings of the First Joint Workshop on Artificial Intelligence and Operations Research*, Timberline Lodge, Oregon, to appear.
- [11] P. Baptiste, C. Le Pape, W.P.M. Nuijten, Satisfiability test and time-bound adjustments for cumulative scheduling problems, Working Paper, Université de Technologie de Compiègne, 1998.
- [12] M. Bartusch, R.H. Möhring, F.J. Radermacher, Scheduling project networks with resource constraints and time windows, *Annals of Operations Research* 16 (1988) 201–240.
- [13] W.W. Bein, J. Kamburowski, M.F.M. Stallmann, Optimal reduction of two-terminal directed acyclic graphs, *SIAM Journal on Computing* 21 (1992) 1112–1129.
- [14] C.E. Bell, J. Han, A new heuristic solution method in resource-constrained project scheduling, *Naval Research Logistic* 38 (1991) 315–331.
- [15] N. Billstein, F.J. Radermacher, Time-cost optimization, *Methods of Operations Research* 27 (1977) 274–294.
- [16] J. Blażewicz, W. Domschke, E. Pesch, The job shop scheduling problem: Conventional and new solution techniques, *European Journal of Operational Research* 93 (1996) 1–33.
- [17] J. Blażewicz, K.H. Ecker, E. Pesch, G. Schmidt, J. Węglarz, *Scheduling Computer and Manufacturing Processes*, Springer, Berlin, 1996.
- [18] J. Blażewicz, J.K. Lenstra, A.H.G. Rinnooy Kan, Scheduling subject to resource constraints, *Discrete Applied Mathematics* 5 (1983) 11–24.
- [19] J. Blażewicz, E. Pesch, M. Sterna, A note on disjunctive graph representation, Working Paper, Universität Bonn, 1998.
- [20] J. Blażewicz, E. Pesch, M. Sterna, A branch and bound algorithm for the job shop scheduling problem, in: A. Drexl, A. Kimms (Eds.), *Beyond Manufacturing Resource Planning (MRP II) – Advanced Models and Methods for Production Planning*, Springer, Berlin, 1998, pp. 219–244.
- [21] J. Böttcher, A. Drexl, R. Kolisch, F. Salewski, Project scheduling under partially renewable resource constraints, *Management Science*, under review.
- [22] F.F. Boctor, Some efficient multi-heuristic procedures for resource-constrained project scheduling, *European Journal of Operational Research* 49 (1990) 3–13.
- [23] F.F. Boctor, A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes, *European Journal of Operational Research* 90 (1996) 349–361.
- [24] K. Bouleimen, H. Lecocq, A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem, Technical Report, Service de Robotique et Automatisation, Université de Liège, 1998.
- [25] R.A. Bowman, Stochastic gradient-based time-cost tradeoffs in PERT networks using simulation, *Annals of Operations Research* 53 (1994) 533–551.
- [26] K. Brinkmann, K. Neumann, Heuristic procedures for resource-constrained project scheduling with minimal and maximal time lags: The resource-leveling and minimum project duration problems, *Journal of Decision Systems* 5 (1996) 129–155.
- [27] P. Brucker, *Scheduling Algorithms*, Springer, Berlin, 2nd edition, 1998.
- [28] P. Brucker, B. Jurisch, A new lower bound for the job-shop scheduling problem, *European Journal of Operational Research* 64 (1993) 156–167.
- [29] P. Brucker, B. Jurisch, A. Krämer, The job-shop problem and immediate selection, *Annals of Operations Research* 50 (1996) 73–114.
- [30] P. Brucker, B. Jurisch, B. Sievers, A branch and bound algorithm for the job-shop problem, *Discrete Applied Mathematics* 49 (1994) 107–127.
- [31] P. Brucker, S. Knust, Solving large-sized resource-constrained project scheduling problems; in: J. Węglarz (Ed.), *Handbook on Recent Advances in Project Scheduling*, Kluwer Academic Publishers, Dordrecht, to appear.
- [32] P. Brucker, S. Knust, A. Schoo, O. Thiele, A branch and bound algorithm for the resource-constrained project scheduling problem, *European Journal of Operational Research*, 107 (1998) 272–288.
- [33] A.R. Burgess, J.B. Killebrew, Variation in activity level on a cyclical arrow diagram, *Journal of Industrial Engineering* 13 (1962) 76–83.

- [34] J.M. Burt, Jr., M.B. Garman, Contitional Monte Carlo: A simulation technique for stochastic network analysis, *Management Science* 18 (1971) 207–217.
- [35] J. Carlier, The one machine sequencing problem, *European Journal of Operational Research* 11 (1982) 42–47.
- [36] J. Carlier, E. Pinson, An algorithm for solving the job-shop problem, *Management Science* 35 (1989) 164–176.
- [37] J. Carlier, E. Pinson, A practical use of Jackson's preemptive schedule for solving the job shop problem, *Annals of Operations Research* 26 (1990) 269–287.
- [38] Y. Caseau, F. Laburthe, Disjunctive scheduling with task intervals, Working Paper, Ecole Normale Supérieure Paris, 1995.
- [39] Y. Caseau, F. Laburthe, Cumulative scheduling with task intervals, *Proceedings of the Joint Internal Conference and Symposium on Logic Programming*, 1996.
- [40] N. Christofides, R. Alvarez-Valdés, J.M. Tamarit, Project scheduling with resource constraints: A branch and bound approach, *European Journal of Operational Research* 29 (1987) 262–273.
- [41] E.W. Davis, G.E. Heidorn, An algorithm for optimal project scheduling under multiple resource constraints, *Management Science* 17 (1971) 803–816.
- [42] E.W. Davis, J.H. Patterson, A comparison of heuristic and optimum solutions in resource-constrained project scheduling, *Management Science* 21 (1975) 944–955.
- [43] N. Dayanand, R. Padman, Payments in projects: A contractor's model, Working Paper, The Heinz School, Carnegie Mellon University, Pittsburgh, 1993.
- [44] N. Dayanand, R. Padman, Project contracts and payment schedules: The client's problem, Working Paper, The Heinz School, Carnegie Mellon University, Pittsburgh, 1996.
- [45] P. De, E.J. Dunne, J.B. Ghosh, C.E. Wells, The discrete time–cost tradeoff problem revisited, *European Journal of Operational Research* 81 (1995) 225–238.
- [46] P. De, E.J. Dunne, J.B. Ghosh, C.E. Wells, Complexity of the discrete time–cost tradeoff problem for project networks, *Operations Research* 45 (1997) 302–306.
- [47] E. Demeulemeester, Minimizing resource availability costs in time-limited project networks, *Management Science* 41 (1995) 1590–1598.
- [48] E. Demeulemeester, W. Herroelen, A branch-and-bound procedure for the multiple resource-constrained project scheduling problem, *Management Science* 38 (1992) 1803–1818.
- [49] E. Demeulemeester, W. Herroelen, New benchmark results for the resource-constrained project scheduling problem, *Management Science* 43 (1997) 1485–1492.
- [50] E. Demeulemeester, W. Herroelen, S.E. Elmaghraby, Optimal procedures for the discrete time/cost trade-off problem in project networks, *European Journal of Operational Research* 88 (1996) 50–68.
- [51] B. De Reyck, Scheduling projects with generalized precedence relations: Exact and heuristic procedures, Ph.D. Dissertation, Katholieke Universiteit Leuven, 1998.
- [52] B. De Reyck, E. Demeulemeester, W. Herroelen, Local search methods for the discrete time/resource trade-off problem in project networks, Technical Report, Department of Applied Economics, Katholieke Universiteit Leuven, 1997.
- [53] B. De Reyck, W. Herroelen, The multi-mode resource-constrained project scheduling problem with generalized precedence constraints, Working Paper, Katholieke Universiteit Leuven, 1997.
- [54] B. De Reyck, W. Herroelen, A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations, *European Journal of Operational Research*, to appear.
- [55] L.P. Devroye, Inequalities for the completion times of stochastic PERT networks, *Mathematics of Operations Research* 4 (1979) 441–447.
- [56] B. Dodin, Bounding the project completion time distribution in PERT networks, *Operations Research* 33 (1985) 862–881.
- [57] U. Dorndorf, E. Pesch, T. Phan Huy, A time-oriented branch-and-bound algorithm for project scheduling with generalized precedence constraints, Working Paper, Universität Bonn, 1998.
- [58] U. Dorndorf, T. Phan Huy, E. Pesch, A survey of interval capacity consistency tests for time- and resource-constrained scheduling, in: J. Węglarz (Ed.), *Handbook on Recent Advances in Project Scheduling*, Kluwer Academic Publishers, Dordrecht, 1998, to appear.
- [59] P.J. Downey, Distribution-free bounds on the expectation of the maximum with scheduling applications, *Operations Research Letters* 9 (1990) 189–201.
- [60] A. Drexl, Scheduling of project networks by job assignment, *Management Science* 37 (1991) 1590–1602.
- [61] A. Drexl, J. Grünewald, Nonpreemptive multi-mode resource-constrained project scheduling, *IIE Transactions* 25 (5) (1993) 74–81.
- [62] S.M. Easa, Resource leveling in construction by optimization, *Journal of Construction Engineering and Management* 115 (1989) 302–316.
- [63] S.E. Elmaghraby, On the expected duration of PERT type networks, *Management Science* 13 (1967) 299–306.
- [64] S.E. Elmaghraby, *Activity Networks – Project Planning and Control by Network Models*, Wiley, New York, 1977.
- [65] S.E. Elmaghraby, The estimation of some network parameters in the PERT model of activity networks: Review and critique, in: R. Slowiński, J. Węglarz, (Eds.), *Advances in Project Scheduling*, Elsevier, Amsterdam, 1989, pp. 371–432.
- [66] S.E. Elmaghraby, Resource allocation via dynamic programming in activity networks, *European Journal of Operational Research* 88 (1992) 50–86.
- [67] S.E. Elmaghraby, Activity nets: a guided tour through some recent developments, *European Journal of Operational Research* 82 (1995) 383–408.
- [68] S.E. Elmaghraby, W. Herroelen, The scheduling of activities to maximize the net present value of projects,



- European Journal of Operational Research 49 (1990) 35–49.
- [69] S. Foldes, F. Soumis, PERT and crashing revisited: Mathematical generalizations, *European Journal of Operational Research* 64 (1993) 286–294.
- [70] B. Franck, K. Neumann, Resource-constrained project scheduling with time windows – structural questions and priority-rule methods, Report WIOR-492, Universität Karlsruhe, 1997.
- [71] H. Frank, I.T. Frisch, R. van Slyke, W.S. Chou, Optimal design of centralized computer networks, *Networks* 1 (1970) 43–58.
- [72] E.C. Freuder, A sufficient condition of backtrack-free search, *Journal of the ACM* 29 (1982) 24–32.
- [73] E.C. Freuder, A sufficient condition for backtrack-bounded search, *Journal of the ACM* 32 (1985) 755–761.
- [74] D.R. Fulkerson, A network flow computation for project cost curves, *Management Science* 7 (1961) 167–178.
- [75] D.R. Fulkerson, Expected critical path lengths in PERT networks, *Operations Research* 10 (1962) 808–817.
- [76] W. Gaul, On stochastic analysis of project-networks, in: M.A.H. Dempster, J.K. Lenstra, A.H.G. Rinnoy Kan (Eds.), *Deterministic and Stochastic Scheduling*, Reidel, Dordrecht, 1982, pp. 297–309.
- [77] R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling theory: A survey, *Annals of Discrete Mathematics* 5 (1979) 287–326.
- [78] R.C. Grinold, The payment scheduling problem, *Naval Research Logistics Quarterly* 19 (1972) 123–136.
- [79] A.V. Goldberg, R.E. Tarjan, A new approach to the maximum-flow problem, *Journal of the Association for Computing Machinery* 35 (1988) 921–940.
- [80] J.N. Hagstrom, Computational complexity of PERT problems, *Networks* 18 (1988) 139–147.
- [81] J.N. Hagstrom, Computing the probability distribution of project duration in a pert network, *Networks* 20 (1990) 231–244.
- [82] R.B. Harris, *Precedence and Arrow Networking Techniques for Construction*, Wiley, New York, 1978.
- [83] R.B. Harris, Packing method for resource leveling (pack), *Journal of Construction Engineering and Management* 116 (1990) 39–43.
- [84] S. Hartmann, Project scheduling with multiple modes: A genetic algorithm, *Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel*, no. 435, 1997.
- [85] S. Hartmann, A competitive genetic algorithm for resource-constrained project scheduling, *Naval Research Logistics*, to appear.
- [86] S. Hartmann, A. Drexl, Project scheduling with multiple modes: A comparison of exact algorithms, *Networks*, to appear.
- [87] R.T. Harvey, J.H. Patterson, An implicit enumeration algorithm for the time/cost tradeoff problem in project network analysis, *Found. Control Eng.* 4 (1979) 107–117.
- [88] R. Heilmann, A branch-and-bound procedure for MRCPSP/max, Report WIOR-512, Universität Karlsruhe, 1998.
- [89] R. Heilmann, C. Schwindt, Lower bounds for RCPSP/max, Report WIOR-511, Universität Karlsruhe, 1997.
- [90] U. Heller, On the shortest overall duration in stochastic project networks, *Methods of Operations Research* 42 (1981) 85–104.
- [91] P. van Hentenryck, *Constraint Satisfaction in Logic Programming*, MIT Press, Cambridge, MA, 1989.
- [92] W. Herroelen, E. Demeulemeester, B. De Reyck, Resource-constrained project scheduling – A survey of recent developments, *Computers and Operations Research*, to appear.
- [93] W. Herroelen, E.L. Demeulemeester, P. Van Dommelen, An optimal recursive search procedure for the deterministic unconstrained max-npv project scheduling problem, Research Report 9603, Department of Applied Economics, Katholieke Universiteit Leuven, 1996.
- [94] W. Herroelen, E. Demeulemeester, B. De Reyck, A classification scheme for project scheduling problems, Technical Report, Department of Applied Economics, Katholieke Universiteit Leuven, 1997.
- [95] W. Herroelen, P. Van Dommelen, E. Demeulemeester, Project network models with discounted cash flows: A guided tour through recent developments, *European Journal of Operational Research* 100 (1997) 97–121.
- [96] T.J. Hindelang, J.F. Muth, A dynamic programming algorithm for Decision CPM networks, *Operations Research* 27 (1979) 225–241.
- [97] J.A. Hoogeveen, J.K. Lenstra, S.L. Van de Velde, Sequencing and scheduling, in: M. Dell’Amico, F. Maffioli, S. Martello (Eds.), *Annotated Bibliographies in Combinatorial Optimization*, Wiley, New York, 1997, pp. 181–197.
- [98] O. Icmeli, S.S. Erengüç, A branch and bound procedure for the resource constrained project scheduling problem with discounted cash flows, *Management Science* 42 (1996) 1395–1408.
- [99] G. Igelmund, F.J. Radermacher, Preselective strategies for the optimization of stochastic project networks under resource constraints, *Networks* 13 (1983) 1–28.
- [100] G. Igelmund, F.J. Radermacher, Algorithmic approaches to preselective strategies for stochastic scheduling problems, *Networks* 13 (1983) 29–48.
- [101] T.J.R. Johnson, An algorithm for the resource-constrained project scheduling problem, Ph.D. Dissertation, MIT, Boston, USA, 1967.
- [102] J.E. Kelley, Critical path planning and scheduling: Mathematical basis, *Operations Research* 9 (1961) 296–320.
- [103] J.E. Kelley, The critical path method: Resource planning and scheduling, in: J.F. Muth, G.L. Thompson (Eds.),

- Industrial Scheduling, Prentice Hall, NJ, 1963, pp. 347–365.
- [104] J.E. Kelley, M.R. Walker, *Critical Path Planning and Scheduling: An Introduction*, Mauchly Associates, Ambler, PA, 1959.
- [105] R. Klein, A. Scholl, Computing lower bounds by destructive improvement – An application to resource-constrained project scheduling, Working Paper, Technische Universität Darmstadt, 1997, *European Journal of Operational Research*, to appear.
- [106] W.K. Klein Haneveld, Robustness against dependence in PERT: An application of duality and distributions with known marginals, *Mathematical Programming Study* 27 (1986) 153–182.
- [107] G.B. Kleindorfer, Bounding distributions for a stochastic acyclic network, *Operations Research* 19 (1971) 1586–1601.
- [108] U. Kuhlhorn, H. Schmeck, K. Haase, Experiences with fine-grained parallel genetic algorithms, *Annals of Operations Research*, to appear.
- [109] R. Kolisch, *Project Scheduling under Resource Constraints – Efficient Heuristics for Several Problem Classes*, Physica, Heidelberg, 1995.
- [110] R. Kolisch, A. Drexl, Adaptive search for solving hard project scheduling problems, *Naval Research Logistics* 43 (1996) 23–40.
- [111] R. Kolisch, A. Drexl, Local search for nonpreemptive multi-mode resource-constrained project scheduling, *IIE Transactions* 29 (1997) 987–999.
- [112] R. Kolisch, S. Hartmann, Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis, in: J. Węglarz (Ed.), *Handbook on Recent Advances in Project Scheduling*, Kluwer Academic Publishers, Dordrecht, 1998, to appear.
- [113] R. Kolisch, R. Padman, An integrated perspective of project scheduling, *Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel*, no. 463, 1997.
- [114] R. Kolisch, C. Schwindt, A. Sprecher, Benchmark instances for project scheduling problems; in: J. Węglarz (Ed.): *Handbook on Recent Advances in Project Scheduling*, Kluwer Academic Publishers, Dordrecht, 1998, to appear.
- [115] R. Kolisch, A. Sprecher, PSPLIB – A project scheduling problem library, *European Journal of Operational Research* 96 (1996) 205–216.
- [116] R. Kolisch, A. Sprecher, A. Drexl, Characterization and generation of a general class of resource-constrained project scheduling problems, *Management Science* 41 (1995) 1693–1703.
- [117] B. Korte, R.H. Möhring, Transitive orientation of graphs with side constraints, in: H. Noltemeier (Ed.), *Proceedings of the 11th International Workshop on Graph Theoretical Concepts in Computer Science*, Trauner, Linz, 1985, pp. 143–160.
- [118] A. Krämer, Branch and bound methods for scheduling problems with multiprocessor tasks on dedicated processors, *OR Spektrum* 19 (1997) 219–227.
- [119] V.J. Leon, B. Ramamoorthy, Strength and adaptability of problem-space based neighborhoods for resource-constrained scheduling, *OR Spektrum* 17 (1995) 173–182.
- [120] C. Le Pape, Implementation of resource constraints in ILOG SCHEDULE – A library for the development of constraint-based scheduling systems, *Intelligent Systems Engineering* 3 (1994) 55–66.
- [121] A. Ludwig, R.H. Möhring, F. Stork, A computational study on bounding the makespan distribution in stochastic project networks, Technical Report, Technische Universität Berlin, Fachbereich Mathematik, Berlin, Germany, 1998.
- [122] A.K. Mackworth, Consistency in networks of relations, *Artificial Intelligence* 8 (1977) 99–118.
- [123] P.D. Martin, D.B. Shmoys, A new approach to computing optimal schedules for the job shop scheduling problem, *Proceedings of the Fifth International IPCO Conference*, 1996.
- [124] I. Meilijson, A. Nadas, Convex majorization with an application to the length of critical paths, *Journal of Applied Probability* 16 (1979) 671–677.
- [125] P. Meseguer, Constraint satisfaction problems: an overview, *AICOM* 2 (1989) 3–17.
- [126] A. Mingozzi, V. Maniezzo, S. Ricciardelli, L. Bianco, An exact algorithm for the resource-constrained project scheduling based on a new mathematical formulation, *Management Science* 44 (1998), 714–729.
- [127] R.H. Möhring, Minimizing costs of resource requirements in project networks subject to a fixed completion time, *Operations Research* 32 (1984) 89–120.
- [128] R.H. Möhring, R. Müller, A combinatorial approach to bound the distribution function of the makespan in stochastic project networks, Technical Report, Technische Universität Berlin, Fachbereich Mathematik, Berlin, Germany, 1998.
- [129] R.H. Möhring, F.J. Radermacher, Substitution decomposition for discrete structures and connections with combinatorial optimization, *Annals of Discrete Mathematics* 19 (1984) 257–356.
- [130] R.H. Möhring, F.J. Radermacher, The order-theoretic approach to scheduling: The deterministic case, in: R. Sowiński, J. Węglarz (eds.), *Advances in Project Scheduling*, Elsevier, Amsterdam, 1989, pp. 29–66.
- [131] R.H. Möhring, F.J. Radermacher, The order-theoretic approach to scheduling: The stochastic case, in: R. Sowiński, J. Węglarz (Eds.), *Advances in Project Scheduling*, Elsevier, Amsterdam, 1989, pp. 497–531.
- [132] R.H. Möhring, F.J. Radermacher, G. Weiss, Stochastic scheduling problems I – General strategies, *Zeitschrift für Operations Research Ser. A* 28 (1984) 193–260.
- [133] R.H. Möhring, F.J. Radermacher, G. Weiss, Stochastic scheduling problems II – Set strategies, *Zeitschrift für Operations Research Ser. A* 29 (1985) 65–104.

- [134] R.H. Möhring, F. Stork, Linear preselective policies for stochastic project scheduling, Technical Report, Technische Universität Berlin, Fachbereich Mathematik, Berlin, Germany, 1998.
- [135] R.H. Möhring, F. Stork, M. Uetz, Resource constrained project scheduling with time windows: A branching scheme based on dynamic release dates, Technical Report, Technische Universität Berlin, Fachbereich Mathematik, Berlin, Germany, 1998.
- [136] U. Montanari, Networks of constraints: Fundamental properties and applications to picture processing, *Information Sciences* 7 (1974) 95–132.
- [137] V. Naumann, Measuring the distance to series-parallelity by path expressions, in: E.W. Mayr, G. Schmidt, G. Tinhofer (Eds.), *Proceedings of the 20th International Workshop on Graph-Theoretic Concepts in Computer Science WG'94*, Lecture Notes in Computer Science, vol. 903, Springer, Berlin, 1995, pp. 269–281.
- [138] K. Neumann, C. Schwindt, Projects with minimal and maximal time lags: construction of activity-on-node networks and applications, Report WIOR-447, Universität Karlsruhe, 1995.
- [139] K. Neumann, C. Schwindt, Activity-on-node networks with minimal and maximal time lags and their application to make-to-order production, *OR Spektrum* 19 (1997) 205–217.
- [140] K. Neumann, J. Zhan, Heuristics for the minimum project-duration problem with minimal and maximal time-lags under fixed resource constraints, *Journal of Intelligent Manufacturing* 6 (1995) 145–154.
- [141] K. Neumann, J. Zimmermann, Resource leveling for projects with schedule-dependent time windows, *European Journal of Operational Research*, to appear.
- [142] K. Neumann, J. Zimmermann, Methods for resource-constrained project scheduling with regular and nonregular objective functions and schedule-dependent time windows; in: J. Węglarz (Ed.), *Handbook on Recent Advances in Project Scheduling*, Kluwer Academic Publishers, Dordrecht, 1998, to appear.
- [143] H. Nübel, A branch-and-bound procedure for the resource investment problem with generalized precedence constraints, Report WIOR-516, Universität Karlsruhe, 1998.
- [144] H. Nübel, C. Schwindt, A classification of shifts, schedules, and objective functions in project scheduling, Report WIOR-509, Universität Karlsruhe, 1997.
- [145] W.P.M. Nuijten, Time and Resource Constrained Scheduling: A Constraint Satisfaction Approach, Ph.D. Thesis, Eindhoven University of Technology, 1994.
- [146] W.P.M. Nuijten, E.H.L. Aarts, A computational study of constraint satisfaction for multiple capacitated job-shop scheduling, *European Journal of Operational Research* 90 (1996) 269–284.
- [147] W.P.M. Nuijten, C. Le Pape, Constraint based job shop scheduling with ILOG SCHEDULER, *Journal of Heuristics* 3 (1998) 271–286.
- [148] O. Oguz, H. Bala, A comparative study of computational procedures for the resource constrained project scheduling problem, *European Journal of Operational Research* 72 (1994) 406–416.
- [149] L. Özdamar, A genetic algorithm approach to a general category project scheduling problem, Research Report, Marmara University, Istanbul, 1996.
- [150] L. Özdamar, G. Ulusoy, A local constraint based analysis approach to project scheduling under general resource constraints, *European Journal of Operational Research* 79 (1994) 287–298.
- [151] L. Özdamar, G. Ulusoy, A survey on the resource-constrained project scheduling problem, *IIE Transactions* 27 (1995) 574–586.
- [152] R. Padman, D.E. Smith-Daniels, V.L. Smith-Daniels, Heuristic scheduling of resource-constrained projects with cash flows: An optimization approach, *Naval Research Logistics Quarterly* 44 (1997) 365–381.
- [153] J.H. Patterson, Project scheduling: The effect of problem structure on heuristic performance, *Naval Research Logistics Quarterly* 23 (1976) 95–124.
- [154] J.H. Patterson, A comparison of exact approaches for solving the multiple constrained resource project scheduling problem, Research Report, Department of Industrial and Manufacturing Systems Engineering, Lehigh University, 1984.
- [155] J.H. Patterson, R. Sowiński, F.B. Talbot, J. Węglarz, An algorithm for a general class of precedence and resource constrained scheduling problems, in: R. Sowiński, J. Węglarz (Eds.), *Advances in Project Scheduling*, Elsevier, Amsterdam, 1989, pp. 3–28.
- [156] J.H. Patterson, R. Sowiński, F.B. Talbot, J. Węglarz, Computational experience with a backtracking algorithm for solving a general class of resource constrained scheduling problems, *European Journal of Operational Research* 90 (1990) 68–79.
- [157] E. Pesch, *Learning in Automated Manufacturing*, Physica, Heidelberg, 1994.
- [158] E. Pesch, U. Tetzlaff, Constraint propagation based scheduling of job shops, *INFORMS Journal on Computing* 8 (1996) 144–157.
- [159] S. Phillips, Jr., M.I. Dessouky, Solving the project time/cost tradeoff problem using the minimal cut concept, *Management Science* 24 (1977) 393–400.
- [160] S. Phillips, Jr., M.I. Dessouky, The cut search algorithm with arc capacities and lower bounds, *Management Science* 25 (1979) 396–404.
- [161] J.S. Provan, M.O. Ball, The complexity of counting cuts and of the probability that a graph is connected, *SIAM Journal on Computing* 12 (1983) 777–788.
- [162] F.J. Radermacher, Invarianzaussagen für stochastische Netzpläne, *Methods of Operations Research* 22 (1976) 136–148.
- [163] P. Robillard, M. Trahan, Expected completion time in PERT networks, *Operations Research* 24 (1976) 177–182.

- [164] D.R. Robinson, A dynamic programming solution to the cost–time tradeoff for CPM, *Management Science* 22 (1975) 158–166.
- [165] B. Rothfarb, H. Frank, D.M. Rosenbaum, K. Steiglitz, D.J. Kleitman, Optimal design of offshore natural-gas pipeline systems, *Operations Research* 18 (1970) 992–1020.
- [166] A.H. Russel, Cash flows in networks, *Management Science* 16 (1970) 357–373.
- [167] R.A. Russell, A comparison of heuristics for scheduling projects with cash flows and resource restrictions, *Management Science* 32 (1986) 1291–1300.
- [168] F. Salewski, A. Schirmer, A. Drexler, Project scheduling under resource and mode identity constraints: Model, complexity, methods, and application, *European Journal of Operational Research* 102 (1997) 88–110.
- [169] S.E. Sampson, E.N. Weiss, Local search techniques for the generalized resource constrained project scheduling problem, *Naval Research Logistics* 40 (1993) 665–675.
- [170] D. Savin, S. Alkass, P. Fazio, A procedure for calculating the weight-matrix of a neuronal network for resource leveling, *Advances in Engineering Software* 28 (1997) 277–283.
- [171] A. Schirmer, A. Drexler, Allocation of partially renewable resources – concept, models and application, *Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel*, no. 455, 1997.
- [172] A. Schoo, Untere Schranken und Immediate Selection für das Resource-Constrained-Projekt Scheduling Problem, *Diplomarbeit, Fachbereich Mathematik/Informatik, Universität Osnabrück*, 1996.
- [173] C. Schwindt, Generation of resource-constrained scheduling problems with minimal and maximal time lags, *Report WIOR-489, Universität Karlsruhe*, 1996.
- [174] C. Schwindt, Verfahren zur Lösung des ressourcenbeschränkten Projektdauerminimierungsproblems mit planungsabhängigen Zeitfenstern, *Shaker, Aachen*, 1998.
- [175] A.W. Shogan, Bounding distributions for a stochastic PERT network, *Networks* 7 (1977) 359–381.
- [176] C.E. Sigal, A.A.B. Pritsker, J.J. Solberg, The use of cutset in Monte Carlo analysis of stochastic networks, *Mathematics and Computers in Simulation* 21 (1979) 379–384.
- [177] M. Skutella, Approximation algorithms for the discrete time–cost tradeoff problem, in: *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, New Orleans, LA, 1997, pp. 501–508.
- [178] M. Skutella, Approximation and randomization in scheduling, *Ph.D. Thesis, Technische Universität Berlin, Fachbereich Mathematik, Berlin, Germany*, 1998.
- [179] R. Sowiński, B. Sońiewicki, J. Węglarz, DSS for multi-objective project scheduling subject to multiple-category resource constraints, *European Journal of Operational Research* 79 (1994) 220–229.
- [180] R.M. van Slyke, Monte Carlo methods and the PERT problem, *Operations Research* 11 (1963) 839–860.
- [181] H. Soroush, Risk taking in stochastic PERT networks, *European Journal of Operational Research* 67 (1993) 221–241.
- [182] H.M. Soroush, The most critical path in a PERT network, *Journal of the Operational Research Society* 45 (1994) 287–300.
- [183] H.G. Spelde, *Stochastische Netzpläne und ihre Anwendung im Baubetrieb*, Ph.D. Thesis, Rheinisch-Westfälische Technische Hochschule Aachen, 1976.
- [184] A. Sprecher, *Resource-Constrained Project Scheduling – Exact Methods for the Multi-Mode Case*, Springer, Berlin, 1994.
- [185] A. Sprecher, Solving the RCPSP efficiently at modest memory requirements, *Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel*, no. 426, 1996.
- [186] A. Sprecher, A. Drexler, Solving multi-mode resource-constrained project scheduling problems by a simple, general and powerful sequencing algorithm, *European Journal of Operational Research*, 107 (1998) 431–450.
- [187] A. Sprecher, S. Hartmann, A. Drexler, An exact algorithm for project scheduling with multiple modes, *OR Spektrum* 19 (1997) 195–203.
- [188] J.P. Stinson, E.W. Davis, B.M. Khumawala, Multiple resource-constrained scheduling using branch and bound, *AIIE Transactions* 10 (1978) 252–259.
- [189] F. Stork, A branch and bound algorithm for minimizing expected makespan in stochastic project networks with resource constraints, *Technical Report, Technische Universität Berlin, Fachbereich Mathematik, Berlin, Germany*, 1998.
- [190] R.S. Sullivan, J.C. Hayya, A comparison of the method of bounding distributions (MBD) and Monte Carlo simulation for analyzing stochastic acyclic networks, *Operations Research* 28 (1980) 614–617.
- [191] M. Takamoto, N. Yamada, Y. Kobayashi, H. Nonaka, Zero–one quadratic programming algorithm for resource leveling of manufacturing process schedules, *Systems and Computers in Japan* 26 (1995) 68–76.
- [192] F.B. Talbot, Resource-constrained project scheduling with time–resource tradeoffs: The nonpreemptive case, *Management Science* 28 (1982) 1197–1210.
- [193] F.B. Talbot, J.H. Patterson, An efficient integer programming algorithm with network cuts for solving resource-constrained scheduling problems, *Management Science* 24 (1978) 1163–1174.
- [194] A. Thesen, Heuristic scheduling of activities under resource and precedence restrictions, *Management Science* 23 (1976) 412–422.
- [195] E. Tsang, *Foundations of Constraint Satisfaction*, Academic Press, London, 1993.
- [196] G. Ulusoy, L. Özdamar, A constrained-based perspective in resource constrained project scheduling, *International Journal of Production Research* 32 (1994) 693–705.
- [197] D. Waltz, Understanding line drawings of scenes with shadows, in: P.H. Winston (Ed.), *Psychology of Com-*

- puter Vision, McGraw Hill, Cambridge, MA, 1975, pp. 19–91.
- [198]z G. Weiss, Stochastic bounds on distributions of optimal value functions with applications to PERT, network flows and reliability, *Operations Research* 34 (1986) 595–605.
- [199] K.K. Yang, L.C. Tay, C.C. Sum, A comparison of stochastic scheduling rules for maximizing project net present value, *European Journal of Operational Research* 85 (1995) 327–339.
- [200] M.A. Younis, B. Saad, Optimal resource leveling of multi-resource projects, *Computers and Industrial Engineering* 31 (1996) 1–4.
- [201] J. Zhan, Heuristics for scheduling resource-constrained projects in MPM networks, *European Journal of Operational Research* 76 (1994) 192–205.
- [202] J. Zimmermann, Heuristics for resource-leveling problems in project scheduling with minimum and maximum time lags, Report WIOR-491, Universität Karlsruhe, 1997.
- [203] J. Zimmermann, H. Engelhardt, Lower bounds and exact algorithms for resource levelling problems, Report WIOR-517, Universität Karlsruhe, 1998.