New MILP Models for Multi-mode Time Constrained Project Scheduling with Temporal Constraints Applied to the Aeronautical Industry

Tamara Borreguero Sanchidrián

Submission date: June 2013
Responsible professors: Álvaro García Álvarez - Miguel Ortega Mier

U.D. Ingeniería de Organización y Logística
Dpto de Ing. Organización, Administración de Empresas y Logística
ETSII - Universidad Politécnica de Madrid

Resumen Ejecutivo

La industria aeronáutica ha sufrido, en los últimos veinte años, un cambio profundo en su contexto industrial. Debido a la entrada en el mercado civil de nuevos actores y unido a la reducción del gasto militar que acompañó el final de la guerra fría, la eficiencia y la competitividad han ganado importancia para la permanencia en el sector. Los sistemas de producción lean y la estandarización se han utilizado masivamente como método de mejora continua. Dentro de este contexto y debido a su impacto en todos los ámbitos de la empresa, la mejora de la planificación a todos los niveles es un facilitador fundamental. Por tanto, ser capaz de producir una planificación de detalle factible y exacta se ha convertido en prioritario.

Además, el secuenciado de tareas con recursos limitados es un problema de optimización NP-complejo. Más aún, es uno de los problemas más difíciles de tratar. Tanto por su relevancia industrial como por su dificultad técnica, la resolución de problemas de secuenciado con recursos limitados está siendo objeto de numerosas investigaciones. A pesar de esto, los problemas de secuenciado que suelen ser estudiados en la literatura no cubren todas las características de muchos problemas reales, entre los que se encuentra el secuenciado de tareas dentro de las plataformas de montaje aeronáuticas.

En este trabajo se ha utilizado la programación lineal basada en eventos para proponer dos nuevas formulaciones para el problema de secuenciado de actividades multimodo en un entorno de tiempo limitado. Se trata de formulaciones de aplicabilidad directa a casos reales de diferentes industrias, entre las que se encuentra la industria aeronáutica. Las principales contribuciones con respecto a las formulaciones existentes se centran en el hecho de contemplar actividades que pueden ser realizadas en varios modos y, a la vez, el secuenciado con un horizonte de tiempo limitado e incluyendo diferentes tipos de relaciones de precedencias entre las actividades.

Junto con estas nuevas formulaciones, se han generado nuevas instancias de datos que completan las existentes en la literatura. Estas instancias nos han posibilitado realizar un estudio computacional, del cual se ha obtenido la comparación de las dos formulaciones propuestas. Finalmente, se han desarrollado nuevos indicadores que permitirán caracterizar las instancias de datos reales, facilitando una futura optimización de los modelos y de su tiempo de resolución.

Abstract

The aeronautical industry has suffered on the last twenty years a significant change in its paradigma. Efficiency has became a major issue due to the increase of competitors and the reduction in military budgets that came after the end of the cold war. Lean production systems and standarization are being used to improve the industry performance. Scheduling and production planning have happened to be on the basis of the achievable improvements in most of a production plant's departments including not only production but also procurement, logistics and other support teams. Therefore, being able to deliver an accurate schedule has became a priority.

At the same time, the scheduling of work tasks with scarce resources is an NP Hard optimization problem and is actually one of the most intractable classical problems. For both its industrial relevance and its challenging difficulty, solving resource constraint scheduling problems is a flourishing research theme. However, the standard Resource Constraint Scheduling Problem usually does not cover all the characteristics of real world problems, including the scheduling of aeronautical assembly lines.

In this work, we present two new event based mixed integer linear programming formulations for a multimode time constraint scheduling problem of direct application for some industries, including aeronautical assembly lines. Its main contributions are the treatment of general precedence constraints, the possibility of having activities with multiple modes and the use of a time constraint approach instead of a resource constraint one.

Together with the new formulations we present a set of instances that has been used for a computational study and comparison of both models. Finally, we have also developped indicators that can help us characterize different data instances for our problem and will be useful for choosing future performace improvement techniques.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the last years, the continuous changes on every industry have forced enterprises to explore new manufacturing methods in order to comply with the *On time, On Quality, On Cost* (OTOQOC) paradigm . Production systems based on the Toyota Production System have spread worldwide as a means of reducing waste and optimizing manufacturing processes.

The aeronautical industry, since the 1990's has been including the lean techniques into its production systems. In terms of Boeing, in a Lean production system the right resources and the right tools must be applied to achieve three key Lean principles: Takt Paced Production, One Piece Flow and Pull Production [Gas02]. Scheduling and line balancing have therefore become two main enablers for lean implementation.

Aeronautical Final Assembly Lines consist on different platforms or stations. Each platform has a fixed team of workers with different skills. The line balancing enables the distribution of the work tasks among the different platforms. Afterwards, the work tasks from each platform must be scheduled in order to complete them within the required Takt Time using the minimum number of operators. We will refer to the scheduling of the tasks as the Aeronautical Platform Scheduling Problem (APSP).

This detailed scheduling has the structure of a Time Constrained Scheduling Problem (TCSP): a project scheduling where a deadline must be met and the objective is to use as few resources as possible. This problem has been very little dealt with in the litterature. Although Möhring introduced it in 1984 [Moh84], only two recent references have been found: [TG08] and [HKPS11]. However, it is a special case of the Resource Constrained Scheduling Problem (RCSP), which was defined by Brucker[BDM$^+$99] as the allocation of scarce resources to dependent activities over time. It is a NP Hard optimization problem and is actually one of the most intractable classical problems.

There have been a wide range of studies on both heuristic and metaheuristic methods for solving the RCSP, as well as different Mixed Integer Linear Programming (MILP) models. Recently, Kone [KALM11] proposed the use of Event Based Formulations for the RCSP. He provided a benchmark of different methods (including MILP exact methods and an heuristic) and concluded that event based formulations outperformed the previous MILP models and performed even better than the heuristic for some instances.

However, Kone's Event Based Formulations deals with the standard RCSP, which includes some assumptions that are too restrictive for many practical applications [HB10]. Therefore, it is of great interest to improve this kind of formulations so that they can be used on more practical applications.

On this work, we have developed a new Event Based Formulation that covers the characteristics of an aeronautical Assembly Platform scheduling problem. Actually, its contribution in this sense is threefold. To begin with, it includes the allowance of **multiple modes per task** as well as the use of more **general temporal constraints** and it has been dealt as a **TCSP** rather than a RCSP. This alternative approach, **focused on minimizing the cost**, is more suitable for nowadays aeronautical industries where the total Lead Time is usually fixed by the expected production rate or the client demand.

For the experimentation, we have created a new set of instances of up to 11 tasks each, due to the different structure of the problem with respect to the prevoius existing ones. The computational results have been concluding enough to select the most efficient Event Based Formulation among the two proposed. Also, the major factors that have an impact on the instance hardness have been identified, providing with directions for a further study and algorithm optimization. Another outcome of this experimentation has been a deeper understanding of the impact of different factors on the problem hardness, enabling the establishment of some indicators to classify different sets of data for our problem.

Moreover, all this work sets the foundations for the doctoral thesis development, including the improvement of the model in order to provide results for real-industrial instances.

Methodology

The current work provides an answer to a series of knowledge gaps related to aeronautical platforms' scheduling, identified during my current work within Airbus Military. After a state of the industry research, which confirmed that the needs are common to other aircraft manufacturers, the critical literature review revealed that some of the specific features of our problem had not been previously addressed and are, at the same time, attractive not only for aeronautical industry but of direct application to other sectors.

Once assured the interest of the topic, the work has been developped under the framework of Operations Research (OR). This is, as defined by Lawrence et al [LP02], *"The discipline that adapts the scientific approach for problem solving to executive decision making in order to accomplish the goal of "doing the best you can with what you've got"*. Although this area of research has widened its limits to be defined as *"an approach to problem solving that employs quantitative analysis to help managers make decisions"* [Den91], scheduling problems (like the one addressed here) remains one of the core problems addresed by OR.

Therefore, throughout my work I have gone through the major phases of an OR project [CA57]:

- **Formulating the problem:**At this stage, statements of objectives, constraints on solutions, appropriate assumptions, descriptions of processes, data requirements, alternatives for action and metrics for measuring progress have been introduced.
- **Constructing a mathematical model to represent the system under study:**I have dentified the kind of problem to be addressed within the literature, together with the sort of model suitable for its representation. Among all the available ones, two different models were chosen as starting points.
- **Solving the model:**I have solved the problem by means of exact mixed integer linear programming. To do so, the models have been implemented in AIMMS and solved with CPLEX.
- **Testing the model and the solution derived from it:** A series of experiments were carried out with sample instances in order to proof the goodness of the solutions and the model's performace.
- **Establishing controls over solution:** Conclusions have been driven from the experimentation as for key instance features and their influence on the solution time, as well as future improvement opportunities.
- **Communicating:** This report has not been the only input of this work. The article "A MILP Event Based Formulation for a Real-world Multimode RCSP with Generalized Temporal Constraints" has been admitted in for the CIO 2013. Also, it has led to an innovation project within Airbus Military.

Document Structure

The structure of this document is as follows: it begins with a description of the industrial context of the problem, the aeronautical industry.

A detailed state of the art review on RCSPs is included on chapter 3. It begins with the definition of RCSP and TCSP. Also, some definitions and terms are explained as well as the general classification for RCSP. Moreover, we present review of the different instance classification and indicators. To end with, the different existing solution methods are presented, focusing on MILP and, most of all, on the Event Based Formulations, which have been chosen to develop our model.

As a consequence of the context and literature review, section 3.8 sets the scope of this research.

Chapter 4 presents the Aeronatical Platform Scheduling Problem. The kind of temporal and resource constraints that will be taken into account, as well as other assumptions included on the model are described. Also, this problem is classified according to the general RCSP characterization.

Two new multimode Event Based Formulations are proposed on chapter 5. Taking as a starting point Koné's formulations, it has been improved in order to deal with multimode tasks, general temporal constraints and to a time constraint approach instead of a resource constraint one. The validity of this formulation is discussed on chapter 6. This chapter focuses on the most important features of the model: regularity of the objective function and the minimum number of events.

The model has been implemented using an operations research modeler (AIMMS). Chapter 7 includes the results of this experimentation, as well as the conclusions on the main impact factors on instance hardness. Following this conclusions, a new set of instance classifications is proposed, as a means of anticipation for further research.

To end the document, chapter 8 is a general conclusion on the Aeronautical Platform Scheduling Problem. It summarizes the results achieved up to know. In addition to this, the further research directions in order to complete the doctoral thesis are explanied.

A part of this work has already been admitted on the 7th International Conference on Industrial Engineering and Industrial Management. The document presented for this conference, together with the excellent assessment of the reviewers are included on appendix C.

Chapter 2

Aeronautical Industry Context: Present and Challenges for Scheduling in The Aeronautical Industry

Murman, [EMMR00], provides a overview of the main changes the aeronautical industry has recently come through: from the Second World War to the 1990's, this industry had strongly relied on public investment. The end of the Cold War prompted drastic reductions in defence procurement budgets resulting in reduced military markets. The defence industry could no longer justify the cost-plus mentality that characterised the Cold War era and faced the challenge of seeking new markets.

At the same time, passenger demand fell suddenly following the Gulf War, forcing airlines to cancel or postpone civil aircraft orders. This followed a period where civil aircraft orders had been running at unprecedented high levels. The inability of the industry to respond to unexpected changes in demand was reflected by long lead times.

Third, in common with other industries globalization has become a central feature. The rise of globalization has clearly demanded a complete rethinking for some firms in terms of how they can organize and reconfigure themselves.

In summary: the aeronautical industry faced the challenge to produce products and systems Better, Faster, Cheaper (or, On Time On Quality On Cost). This new paradigm is a considerable change from the mantra of Higher, Faster, Farther that had been the driving force behind aerospace products and systems for many years. [EMMR00]

On its search to adapt to this new paradigm, the aeronautical industry have looked forward to implementing the Toyota Production System (TPS), dubbed the "lean" approach by former MIT student and researcher John Krafcik. That Toyota Production System has been running on Japan since the 1950's. It is a process oriented approach that addresses directly the need for process improvement in the aerospace industry in this era. Lean represents a new way of thinking about organising and executing tasks and activities to achieve project and organizational goals through adding value by eliminating wasteful practices.

Many references have been written about the Lean Principles and Tools. Toyota engineer, Taiichi Ohno, published *Toyota Production System* in 1988, [Ohn88], which outlines Ohno's quest and provides insights into the crucial process of innovation that are valuable for managers of all types. Also, Womack and Dan Jones, founders of the Lean Enterprise Institute [Lea] and the UK Lean Enterprise Academy respectively, wrote *Lean Thinking* in 1996 [JPW96]. That book, turned out to be among the ones with most impact on general industry transformation, boosting lean implementation on a wide range of sectors.

Despite the extension of lean techniques to different industries, there is still in some areas the perception that Lean manufacturing is to some degree, an "automotive idea" and difficult to transfer to other sectors especially when there are major differences between them. Due to this perception, Crute et al [CWBG03] discussed the key drivers for Lean in aerospace and examined the assumption that cross-sector transfer may be difficult. They included a Lean implementation case comparison examining how difficulties that arise may have more to do with individual plant context and management than with sector specific factors.

Nevertheless, both the top two aircraft manufacturers Airbus and Boeing have made a huge effort on Lean implementation during the last decades. Boeing began in 1995 with an ambitious benchmarking where managers studied up to eight Japanese WorldClass companies [LP05]. Lean culture was spread all over the organization and tools and techniques were introduced, begining with statistical process control and variability reduction techniques. Airbus has also included lean principles since the 1990's within its way of working. Making a leaner and more fully integrated company has been consolidated on its 2012 Registration Documents as a key aspect to booster its competitiveness [EAD12].

The Lean implementation within a company needs to be accomplished by a deep cultural change. Oftenly, each company formulates on its own words the main Lean Strategies, as well as the roadmap for Lean implementation. However, the principles remain the same. Spear and Bowen, [Spe99], provided a review that focuses on what they call the Four Rules, which summarizes the tacit knowledge that underlie in the Toyota Production System (TPS) rather than describing the tools deployed for its achievements. These Four rules are:

- **Rule 1** All work shall be highly specified as to content, sequence, timing, and outcome.
- **Rule 2** Every customer supplier connection must be direct, and there must be an unambiguous yes-or-no way to send requests and receive responses.

- **Rule 3** The pathway for every product and service must be simple and direct.
- **Rule 4** Any improvement must be made in accordance with the scientific method, under the guidance of a teacher, at the lowest possible level in the organization.

Guided by this rules, where traditional improvement methodologies focused on reducing the product transformation time, Lean seeks the elimination of all non-value activities, classified on seven catergories: overproduction, inventory, waiting, motion, transportation, rework and over-processing. Ofently, an eigth waste is added: under-utilized talent, as the main asset of TPS companies is their personnel.

1. **Overproduction:** To produce sooner, faster or in greater quantities than the customer needs.
2. **Inventory:** Raw material work in progress or finished goods when no value is being added to it.
3. **Waiting:** People or parts that wait for a life cycle to be completed.
4. **Motion:** Movement of people, parts or machinery wthin a process.
5. **Transportation:**Movement of people, parts or machinery between processes.
6. **Rework:**Non-right the first time repetiotion or correction of a process.
7. **Over-Processing:**Processing beyond the standard required by the customer.

In terms of Boeing, the operational change must be driven by three key Lean principles: Takt Paced Production, One Piece Flow and Pull Production [Gas02].

- **Takt Paced Production** is common to Lean and to the Theory Of Constraints introduced by Goldratt in [GEM84]. It implies that the production rythm adapts to a common pace all the production steps. As well as this, that pace or Takt must be adapted to the client demand, in order to produce just on the required amount. In this way, inventories and stock obsolescence happen at a much lower rate.
- **One Piece Flow** runs for the processing of only one piece at a time, instead of batch production. It requires a more efficient change time from one operation to the other.

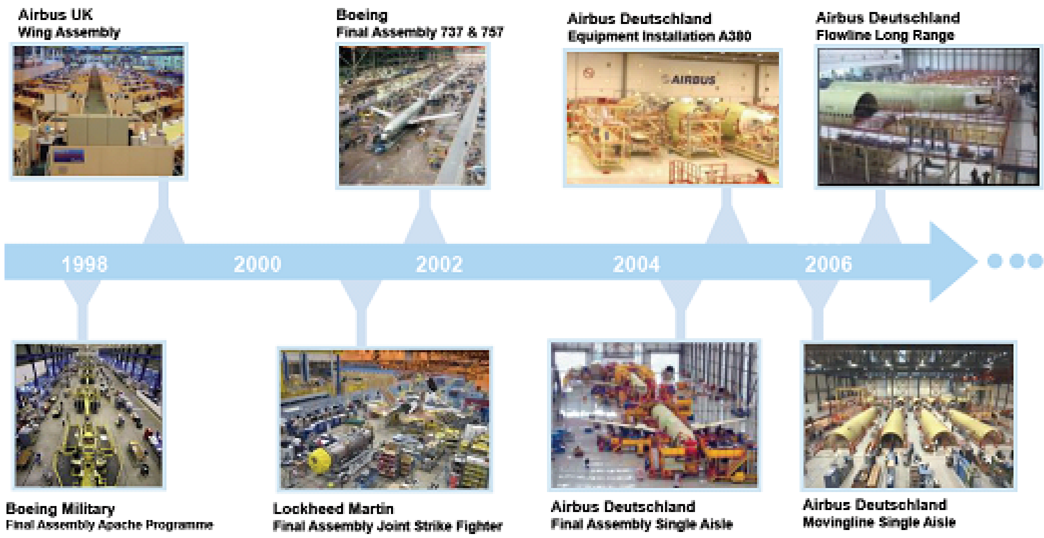

Figure 2.1: Flow Production in Aerospace Industry [Alt10]

- In **Pull Production** , the plant is organized in a way that a production process will not begin until the next step has consumed one unit of the product that it delivers.

Based on this three principles, aeronautical assembly lines have changed from a product orientation, where most of the lines consisted on several platforms where the same jobs were performed in parallel to moving or pulse process oriented lines. In this lines, each of the stages, or platforms produces its output at a rate equal to the rate at which the assembly line produces airplanes. The One Piece Flow principle avoids parallel platforms for the same jobs: the division into two series platforms is prefered (when possible). Due to pull production, the movement of the aircraft between stations is syncronized and determined by the last station.

Examples of this lines have spread all over the aeronautical industry. Some examples can be found on figure 2.1.

The outcome of this moving line implementation has been major improvements in inventory (both in stores and work in progress) and flow time. For example, Boeing

737 line experience a reduction of 46% in stores inventory, 59% in work-in-progress inventory, 55% on factory footprint down . Moreover, the flow time in Final Assembly was 22 days and had turned into 11 days by 2005 [LP05]. Airbus Single Aisle moving line had similar results, which included a flow time reduction from 9 to 5 days and 14 to 8 days on its two products. Those flow time reductions enabled the plant to comply with a rump up from 22 to 32 deliveries per month [Fra07].

This moving line implementation is the final result of a complete transformation. As a guide, Boeing developped *The 9 tactics* , which represent a practical application of Boeing's lean manufacturing concepts, principles and techniques to meet that ultimate goal of implementing a continuous moving line [Gas02]:

- **Tactic 1 - Understand how value flows:** Understanding how value should flow through the business is the first tactic in value creation. At this step Value Stream Mapping is used for the team to gain knowledge on the current state of the business such as flow time, queue time, and the amount of work-in-process inventory.

- **Tactic 2 - Balance the line:** Balancing the line essentially means evenly distributing both the quantity and variety of work across available work time, avoiding overburden and under-use of resources. Work that is evenly distributed provides predictability and the ability to standardize work processes more easily. This eliminates bottlenecks and down time, which translates into shorter flow time.

- **Tactic 3 - Standardize work procedures:** Standard work procedures are the foundation of a Lean production system. A standard operation is a known, repeatable process that results in high-quality output. A standard operation ensures that everyone does the same job in the same way, in the best way possible.

- **Tactic 4 - Put visual controls in place:** Visual controls in the workplace can help people quickly and accurately gauge production status at a glance. These visual systems fall into two basic categories: progress indicators and problem indicators. A production-scheduling scoreboard is an example of a progress indicator.

- **Tactic 5 - Put everything at point of use:** Point of use is a technique that ensures people have exactly what they need to do their jobs the right information, parts, tools, and equipment where and when they need them

- **Tactic 6 - Establish feeder lines:** Feeder lines allow an assembly area to take the preassembly tasks and perform them off the main production line.

These tasks can then be done before or at the same time major assembly tasks are performed on the main line.

- **Tactic 7 - Radically redesign products and business processes:** Breakthrough process redesign uses innovative ideas and concepts to reduce flow time, work in process inventory, and defects.

- **Tactic 8 - Convert to a pulse line:** Transition period in order to put in place reliable processes to respond to potential problems or immediate problems to keep work moving or to quickly resume after work has stopped.

- **Tactic 9 - Convert to a moving line**

The output from tactic 2 is the decision on how many platforms will exist on the line, as well the selections of the works to be done on each platform. On some cases, the assignment of works to a platform is strongly linked with the jigs or tools, as some of them may require a high investment and should not generally be duplicated. Normally, there is another set of works that can be done on either of the platforms. Those will be use in order to balance the line.

Afterwards, for each platform, the different jobs must be scheduled: distributed over time and assigned to the different operators that may work on each platform. This schedule needs to be balanced, as it is the basis not only for standarized work procedures (tactic 3) but also for visual controls and point of use deliveries (tactics 4 and 5).

Aeronautical Assembly lines can gather up to 100000 working hours. Takt times vary from 5 to 60 working days. As a result, the number of works to schedule on a platform can be on the range of 100-500. Therefore, the preparation of these plans is iterative and time-consuming, complicated by constraints such as assembly sequences and space limitations. However, on most of the cases, the process for balancing the line is not standardized and each platform teams prepares the plans based on his/her experience, as is stated for Boeing by [Gas02].

Nevertheless, operational research and simultaion techniques can be effective in improving those barchart solutions, standarising the process and the objectives and reducing the time needed for a result. Normally, the detailed standart production plan can be prepared in advanced and obtaining a good solution is more important than getting a quick one. Further improvements or modifications once the works have started must be quicker and can be less accurate.

They are already some examples for the use of this techniques in detailed aeronautical schedule. For Boeing 747 moving line discrete event simulation was used

[RFL02]. Within Airbus there is already a software that, with an heuristic, proposes a solution.

As time goes by, this schedules must be more effective and accurate, in order to continue with the improvement rates that have been experienced on the last years. They must be able to calculate the minimum needed capacity needed over a time period, and optimally allocate and reoptimize the right resources per shift to the demand. This can only be achieved by new investigation and operational research methods. As an example of this concern, there are already some proyects looking onto this direction. For example, EADS (the consorcium of which Airbus is part) participates, among others, on the ARUM proyect (Adaptative Production Management) [aru], whose objective is to develop special methods for the case of high technological products with low production rates, as is the case of aircrafts.

In conclusion, the aeronautical industry has experienced a deep transformation that has boostered the need for innovation and continuous improvement. Due to its impact on all other areas (human ressources, supply chain management), scheduling is one of the areas where the lack of suitable new methods is more necessary. As a consequence, in next chapter we will review the existing literature on scheduling and the solution methods that have been already proposed.

Chapter 3

State of the Art Review: Resource Constraint Project Scheduling and Master Thesis Scope

3.1 Resource and Time Constraint Project Scheduling

As a first approach, a project scheduling problem consists on a time window and resource assignment for a set activities of known durations and resource requests, that must be executed guaranteeing some precedence relations.

Resources can only be assigned to one task at a time. Hence limited resources may impose additional precedence relations between activities consuming the same resource, thereby possibly increasing the project duration. At the same time, carrying out activities simultaneously in order to save time will usually result in higher costs for the resources consumed.

These considerations lead to the following optimization problems:

- **RCSP - Resource Constrained Scheduling Problem** : Given constant limits for the available amount per resource, its objective is to find the shortest possible project duration.
- **TCSP - Time Constrained Scheduling Problem** : Given a time limit for the project duration the objective is to find the least resource consuming schedule, if resources are assumed to be available in unlimited amounts at a fixed cost.

However, the TCSP can be considered a variant of the RCSP. Although some authors have refered to it separately, [Moh84] already established the paralelism between them in 1984.

Scheduling problems are one type of combinatorial optimization problems. These are defined by a solution space $\mathcal{X}$, which is discrete or which can be reduced to a discrete set, and by a subset of feasible solutions $\mathcal{Y} \subseteq \mathcal{X}$. Each solution is associated with an objective function $f : \mathcal{Y} \to \mathcal{R}$. The aim of the problem is to find a feasible solution $y \in \mathcal{Y}$ such that $f(y)$ is minimized or maximized.

Using the RCSP definition by Artigues [AKL+10], both scheduling problems are combinatorial optimization problem defined by a 6-tuple ($\mathcal{W}$, p , $\mathcal{A}, \mathcal{K}, \mathcal{B}$, p), where:

- $\mathcal{W}$ is a set of activities,
- p is a vector of processing times per activity,
- $\mathcal{A}$ is the set of temporal constraints,
- $\mathcal{R}$ is the set of resources,
- b stands for the demand matrix (resource consumption per activity),
- $\mathcal{B}$ is the resource capacity vector - For the RCSP
- $\mathcal{LT}$ is the lead time capacity vector - For the TCSP

The objective is to identify a feasible schedule, which assigns a start / completion time (S_i/F_i) to each activity as well as a resource allocation, taking into account the temporal constraints and minimizing the total project lead time (RCSP) or the resource consumption (TCSP).

As for the temporal constraints, the most common notation is the activity-on-the-node (AoN) network, where the nodes represent activities and the arcs represent precedence constraints. Information on the processing times per activity can also be added on the graph representation.

Although in practice deadlines often occur in projects, the Time Constrained Project Scheduling has been considered only rarely in the literature. On most of the cases, it has been considered combined with resource contraints, as is the case of scheduling with resource constraints and time windows.

On the rest of the chapter, we will use the term RCSP to refer to the more general case of scheduling problems, in order to use the same notation from the literature. Nevertheless, the conclusions and discusions are valid for the TCSP.

3.2 RCSP Classification and Notation

Until 1999, there was not a common notation for RCSP. Brucker[BDM+99] proposed a notation based on the extension of the $\alpha|\beta|\gamma$ generalized scheme for the machine scheduling and resource constrained machine scheduling literature. In this notation,

α refers to the resource environment, β to the activity characteristics and γ to the objective function.

- α *: Resource Environment.* In order to distinguish project scheduling from machine schedule, the α field will begin with *PS* for project scheduling or *MPS* for the multimode scheduling project case. After them, two sets of parameters refer to renewable (m, σ, ρ) and, for the multi-mode case, non-renewable resources (μ, τ, ω).

  In this way PSm, σ, ρ states for a project scheduling problem with m resources, σ units available per resource and where each activity requires at most ρ units of the resources.

  Similarly, μ represents the number of non-renewable resources, τ the number of units available per non-renewable resource and ω the maximum demand of a non renewable resource per task.

  In consequence, a multi-mode scheduling problem with both renewable and nonrenewable resource will be: $MPSm, \sigma, \rho; \mu, \tau, \omega$. In this notation, whenever $\cdot$ is used instead of an entry, it means that the values of the parameters are to be specified in the input.

  As an example:

  $PS4, 2, 1$ - States for a single mode scheduling problem with 4 resources, 2 units per resource and at most 1 unit of each resource per activity.

  $MPS3, \infty, 3$ - States for a multi-mode scheduling problem with 3 resources, unlimited units per resource and at most a demand of 3 resources per resource type and activity.

- β *: Activity Characteristics.* This parameter is the same as the defined for machine scheduling by Graham et al, [GLLRK77]. Several activity related characteristics are described: $\beta \subset \{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6\}$.

  $\beta_1 = pmtn, \cdot$ Depending on wether preemption is $(pmtn)$ or not $(\cdot)$ allowed.

  $\beta_2 = \{p_w = 1,\ \underline{p}\ ,\ \overline{p}\, p_w = sto\}$ If all procesing times are equal to 1, to a lower $(\underline{p})$ / upper bound $(\overline{p})$or are stochastic.

  $\beta_3 = d, \cdot$ There exists a deadline for the project duration.

  $\beta_4 = prec,\ temp$ If there are general precedence constraints between activities or if that constraints are given by minimum and maximum start-start time lags between activities.

  $\beta_5 = chains,\ intree,\ outtree,\ tree$ When precedence relations are specified by means of chains, intree, outtree, tree.

- γ: *Objective Function.* The objective functions are described using their corresponding formulae. The most classical objective functions are minimizing the project Lead Time (LT) or the tardiness ($LT_{min} - LT_{obj}$), along with the project cost ($\sum B_{min} * C_b$). Another criteria can be the net present value, resource leveling or resource investment.

3.3 Definitions: Tasks Properties and Schedule Classification

On this section we will present some common notation on task properties and schedule classification that will be used from now on.

3.3.1 Task Properties

To begin with, refering to the precedence graph, we will use the set $\mathcal{P}_w(\mathcal{SU}_w) \subset \mathcal{W}$ to denote the inmediate predecessor nodes (sucessor nodes) of activity w by means of an arc. The Sets $\mathcal{P'}_w(\mathcal{SU'}_w)$ are used to denote the sets of all nodes that precede (succeed) node w.

Moreover, given the set of activities, their vector of processing times and the set of temporal constraints it is posible to calculate the problem PERT diagram. In this diagram, we will obtain for each activity the *Earliest Start Date:* ES_w when not taking into account the resource constraints.

If we also define an objective Lead Time, we can calculate the *Latest Start Date:* LS_w per activity as the latest time at which the activity must be started not to impact the objective Lead Time. Therefore, any feasible solution will have a starting time S_w for activity w within the interval:

$$ES_w \leq S_w \leq LS_w \tag{3.1}$$

Once we take into account not only the set of temporal constraints but also the resource constraints, the feasible interval for S_w may be tighter but equation 3.1 will always be a valid cut or starting point for the several RCSP heuristic and exact algorithms. For example, for a five task RCSP, figure 3.1 represents the feasible time windows per task.

This feasible starting times per activity, can also be defined using the Earliest / Latest finish times, EF_w / LF_w :

$$EF_w = ES_w + p_w \tag{3.2}$$

$$LF_w = LS_w + p_w \tag{3.3}$$

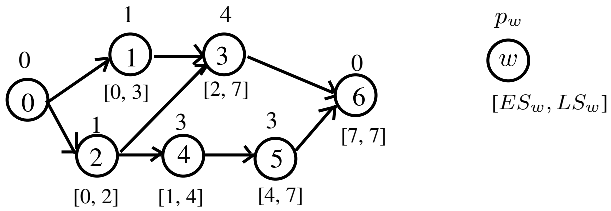

Figure 3.1: Time Windows

Other authors, as [BK11], use the alternative concept of head and tails for each activity:

- **Head:** r_w is a lower bound for the *Earliest Start Date:* $ES_w = r_w$
- **Tail:** q_w is a lower bound for the time between the time period between the completion time of w and the total Lead Time: $q_w = LT - LS_w$

3.3.2 Schedule Classification

Within all the possible schedule solutions for the RCSP, there are different subsets of feasible schedules which are useful to compare different exact and heuristic algorithms. Sprecher, [Spr95] introduced the definition for semi-active, active, and non-delay schedules, being all of them subsets of the feasible schedules.

Given a feasible schedule, we will call a left shift the movement of one activity to the left without increasing (in the case of minimizing objective functions) / decreasing (in the case of maximizing objective functions) the value of the objective schedule. That left shift will transform a feasible schedule $\mathcal{S}$ into another $\mathcal{S}'$ where all the activities $w \neq w_1, w \in \mathcal{W}$ have the same starting time except $t'_{w_1} < t_{w_1}$. In the case that $t'_{w_1} = t_{w_1} - 1$ we will call this a one-period left shift. Sucessive one-period left shifts where all the intermediate left shifts are feasible are called a local left shift. In other cases, an activity can be left shifted more than one time period but some of the intemediate solutions are not feasible. In that case, we will call this movement a global left shift.

Following the local and global left shift definitions, schedules are called semi-active whenever no local left shift is posible for any ot its tasks and active if no local nor global left shift is posible. On both active and semi-active schedules, additional shifts may be posible if we accept preemption of activities, this is: stopping the activity at some point and continuing the processing after an idle period. If no local or global shift is posible even considering preemption the schedule is called a non-delay schedule.

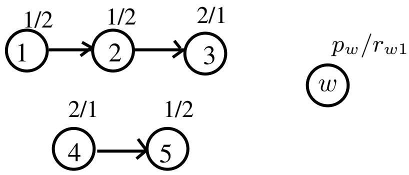

Figure 3.2: RCSP Instance

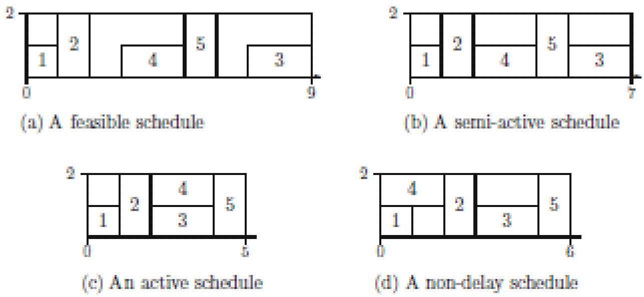

Figure 3.3: RCSP Schedule Classification

As an example, we consider the 5-task problem from figure 3.2, with one resource with constant capacity $R_1 = 2$ [BK11] . Different kind of schedules for this problem are shown on figure 3.3.

Sprecher, [Spr95] studied this schedule subsets and stated that for regular measures, there is always an optimal solution among active schedules. Therefore, any solution method may search only on active shcedules subsets without leaving apart the optimal solution. Semi-active schedules are a wider subset than active schedules. Non-delay schedules do not always include an optimal solution.

3.4 Solution Methods for the RCSP

As other combinatorial problems, the RCSP can be solved using several techniques. Being a NP-Hard problem, instances with more than 60 activities are difficult to solve using exact methods. Therefore, a wide range of both heuristic and metaheuristic methods have been proposed. On this section, we will provide an overview of some of the proposals.

3.4.1 Heuristics

On 1963, Kelley [Kel63] published a first schedule generation heuristic. Since then, a large number of different solution techniques have been proposed. The core of most of them are Schedule Generation Schemes (SGS). In some cases, the schedule generation is enhanced by the use of priority rules, giving birth to a set of heuristics known as priority based RCSP heuristics.

Schedule Generation Schemes

Schedule Generation Schemes (SGS) use a stepwise approach in order to build a feasible schedule, starting by a partial one. This progressive extension of the schedule can be done in two directions: *activity-oriented series SGS* , where one activity is scheduled at each step, and *time-oriented parallel SGS* where at each step, one time moment is considered and several activities may be included on the schedule. [KH99]

On both of the SGS, we will define at each iteration (g) several activity subsets. Subset S_g includes the activities that have already been scheduled. S_g will be divided on two other subsets: $A_g \subset S_g$ stands for the activities which are active at iteration g and $C_g \subset S_g$ for the activities that have already been completed prior to g, where $S_g = (A_g \cup C_g)$. As well as this, D_g, includes the activities that may be eligible for scheduling. Note that D_g may not contain all the not scheduled activities, as some of them can have unsatisfied precedence constraints.

Furthermore, we will define $\tilde{B_k}(t_g) = B_k - \sum_{j \in A_g} b_{wk}$ as the availability of resource k at stage g.

Finally, $F_g = \{F_w | w \in S_g\}$ is the set of finishing times for the scheduled activities.

- *Activity-oriented series SGS:*

  For activity-oriented series SGS, at the beginning of each iteration g, D_g, F_g and $\tilde{B}_k$ are updated. Afterwards, an activity $w \in D_g$ is chosen and it is assigned the earliest resource-feasible finish time F_w within its feasible starting times: $F_w \in [EF_w, LF_w]$. The algorithm description is as follows:

  Activity-oriented series SGSS produce always feasible schedules. Moreover, theY are active schedules, where none of the activities can be started earlier without delaying some other activity. Therefore, , in the case of scheduling problems with a regular performance measure there is always an optimal schedule within the series SGS generated schedules.

- *Time-oriented parallel SGS:*

Algorithm 3.1 Activity-oriented series SGS

Initialization: $F_0 = 0, S_0 = \{\}$
for $g = 1$ *to* n **do**
  Calculate $D_g, F_g, \tilde{B_k}(t)(k \in R, t \in F_g)$
  Select $w \in D_g$
  $EF_w = max_{h \in P_w}\{F_h + p_w\}$
  $F_w = min\{t \in [EF_w - p_w, LF_w - p_w] \cap F_g | b_{wk} \leq \tilde{B}_k, k \in R, \tau \in [t, t + p_w] \cap F_g\} + p_w$
  $S_g = S_{g-1} \cup w$
end for
$F_{n+1} = max_{h \in P_{n+1}}\{F_h\}$

On time-oriented series SGS, for each iteration g, time period is scheduled. On a first step D_g, C_g, A_g and $\tilde{B}_k$ are updated. Afterwards, all the possible activities from D_g are scheduled with finish time $F_w = t_g + p_w$. The algorithm description is as follows:

Algorithm 3.2 Time-oriented parallel SGS

Initialization: $g = 0, t_g = 0, A_0 = 0, C_0 = 0, \tilde{B}_k(0) = B_k\ \forall k \in R$
while $|A_g \cup C_g| \leq n$ **do**
  $g := g + 1$
  $t_g = min_{w \in A_g} F_w$
  Calculate $C_g, A_g, \tilde{B}_k(t_g), D_g$
  while $D_g \neq \emptyset$ **do**
    Select one $w \in D_g$
    $F_w = t_g + p_w$
    Calculate $\tilde{B}_k(t_g), A_g, D_g$
  end while
end while
$F_{n+1} = max_{h \in P_{n+1}} F_h$

The schedules generated by time-oriented parallel SGS are non-delay schedules. As a result, the set of generated schedules may not include an optimal schedule. This is a major drawback of this schedule generation scheme.

Priority Rules

When generating feasible schedules, with either of the SGS, we ofently use priority rules in order to assign to each activity $w \in D_w$ a value $v(w)$ so that the activity with the largest (or smallest) value is chosen. An addional rule must also be defined, to use it in case of ties.

Priority rules can focus on different activity characteristics. The main groups are activity-based, network-based, critial path-based or resource-based rules. Table 3.1 summarizes some of the existing priority rules.

Table 3.1: Existing Priority Rules [BK11]

TYPE	RULE	CHOOSES THE ACTIVITY WITH THE:
Activity-based	SPT	Smallest processing time
Activity-based	LPT	Longest processing time
Network-based	MIS	Most immediate successors
Network-based	LIS	Least immediate successors
Network-based	MTS	Most total successors
Network-based	LTS	Least total successors
Network-based	GRPW	Greatest rank positional weight
Critial path-based	EST	Smallest earliest starting time
Critial path-based	ECT	Smallest earliest completion time
Critial path-based	LST	Smallest latest starting time
Critial path-based	LCT	Smallest latest completion time
Critial path-based	MSLK	Minimum slack
Resource-based	GRR	Greatest resource requirements

If the value v_w assigned to a task remains the same all over the SGS, we say the priority rule is static and it is dynamic otherwise.

Different heuristics are constructed by the combination of one or both SGS and one or more priority rules. The oldest heuristics where *single pass methods*, which employ one priority rule and one SGS in otder to obtain one feasible schedule. *Multi priority rule methods* employ a SGS several times, using different priority rules each time. *Forward-backward scheduling methods* use a SGS to iteratively schedule the project by alternating between forward and backward scheduling. Finally, *sampling methods* use one SGS and a priority rule, but they obtain several schedules by biasing the selection of the priority rule through a random device, as they transform the value v_w into a probability of w being chosen.

Priority-based heuristics are often used in practice since they have small running times and are easy to implement. Furthermore, they are often used to calcute lower/upper bounds or initial solutions.

3.4.2 Metaheuristics

Among all metaheuristics, the most common for the RCSP are genetic algorithms, tabu search, simulated annealing and ant systems.

Genetic Algorithms

First pioneered by John Holland in 1995, [Hol75], Genetic Algorithms (GAs) are based on the evolutionary ideas of natural selection and genetic. GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem.

This algorithms consider a set of feasible schedules, or population. Starting from them new solutions are calculated by mating two existing ones (crossover) or by altering an existing one (mutation). Once the new solutions are produced, the best solutions are chosen, according to the objective function value. The fittest (best) solutions survive, becoming the next generation and the rest are deleted.

It has been one of the most commonly used metaheuristic for our problem. Two examples are those proposed by Hartmann [Har98] and, more recently, an improved algorithm by Valls et al [VV03].

Tabu Search

The basic concept of tabu search was described by Glover ([Glo89a] and [Glo89b]).It is a research method which evaluates all the solutions from a neighbourhood and chooses the best one, in order to proceed from it. This method would have the risk of cycling, going back to the prior situation. In order to avoid this problem, a tabu list is set up as a form of memory for the search process. This list may be overrun only if the corresponding neighbourhood will lead to a new overall best solution (aspiration criterion).

Artigues [AMR03], Klein [Kle00] and Nonobe and Ibaraki [KN02] have proposed some of the most recent tabu search algorithms for the RCSP.

3.4.3 Ant Colony Optimization

Ant colony optimization (ACO) takes inspiration from the foraging behavior of some ant species, [MD96]. These ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony. Ant colony optimization exploits a similar mechanism for solving optimization problems.

In the case of RCSP, each solution is represented by an activity list, which describes the order in which activities have been included in the solution during the SGS. The pheromone value in this case is related with how promising it seems to put activity w as the *ith* activity into the schedule, taking into accout the objective function value for the previous solutions which included that choice.

Merkle et al. [DM6] presented the first ACO approach for a RCSP.

For a more detailed description of several heuristics and metaheuristic methods for solving the RCSP, including computational results, [HK00] and [KH06] may be consulted.

3.5 MILP Formulations for the RCSP

Although up to now only small instances can be solved to optimality by mixed integer linear programming (MILP) methods, they are ofently useful when combined with others in hybrid methods. In fact, most of the best known lower bounds for some common instance sets where obtained by an hybrid method including constraint propagation and MILP formulations. Moreover, in the recent years, more eficient MILP solutions are being presented and therefore this formulations are becoming more promising.

There exists a large number of MILP formulations for our general RCSP. These formulations can be divided on three main groups: Discrete Time Formulations, Continuous Time Formulations and Event Based Formulations. In this section, we will review the first two. Being the Event Based Formulation the starting point for our model, it will be presented on next section.

As a common notation, valid for all of the formulations, we must remember that a feasible schedule will be one that assigns for each task a start time that is feasible given the precedence constraints (3.4) and the resource constraints (3.5)

$$S_{w'} - S_w \geq p_w \ \forall (w, w') \in \mathcal{A} \tag{3.4}$$

$$\sum_{w \in W_t} b_{wk} \leq B_k \ \forall k \in \mathcal{R}, \ \forall t \in T \tag{3.5}$$

Where S_w is the starting time of task w, and $\mathcal{A}$ stands for the precedence relations graph between activities ($(w, w') \in \mathcal{A}$ if w must precede w'). As well as this, W_t

represents the set of non-dummy activities in process at time t. B_k is the total available quantity of the renewable resource $k \in R$ and b_{wk} is the resource demand of type k for activity w.

3.5.1 Discrete Time Formulations

In discrete time formulations, the time horizon T is divided into time slots. The basic discrete time formulation (DT) was proposed by Pritsker [PWW69]. He defined a time-indexed binary variable x_{wt}, where $x_{wt} = 1$ if activity w starts at time t and zero otherwise. As well as this, activity $n + 1$ is the dummy activity for the end of the project.

$$Minimize \sum_{t=0}^{T} t x_{n+1,t} \tag{3.6}$$

Subject to:

$$\sum_{t=0}^{T} x_{wt} = 1 \ \forall w \in \mathcal{W} \tag{3.7}$$

$$\sum_{t=0}^{T} t x_{w't} - \sum_{t=0}^{T} t x_{wt} \geq p_w \ \forall (w, w') \in \mathcal{A} \tag{3.8}$$

$$\sum_{w=1}^{n} b_{wk} - \sum_{\tau=max\{0,t-p_w+1\}}^{t} x_{w\tau} \leq B_k \ \forall t \in (0, T-1) \ \forall k \in R \tag{3.9}$$

$$x_{wt} \in \{0, 1\} \ \forall w \in W, \ t = (0, T) \tag{3.10}$$

Where equation 3.6 is the objective: to minimize the project competion time. Equation 3.7 assures that each activity is programmed once and only once. Precedence relations are satisfied as per equation 3.8. The resource constraints are guaranteed by 3.9, since an activity w is only being processed in the interval $[t, t + 1]$ if and only if its starting time is in the interval $[t - p_w + 1, t]$.

This formulation consists of $O(nT)$ binary variables and $O(|E| + |R|T + n)$ constraints. The number of variables can be reduced if time windows are defined per activity, by the calculation of its earlies / latest start time. In that case, we would define $x_{wt} = 0 \ \forall t \in (0, ES_w) \cup (LS_w, T)$ and the number of variables would be $\sum_{w=1}^{n+1} (LS_w - ES_w)$.

Afterwards, Christofides [Chr87] proposed the Disaggregated Discrete Time formulation (DDT). This formulation replaces the precedence constraints 3.8 by a new set 3.11. On the one hand, it implies a larger number of constraints but, on the other hand, is a tighter formulation and therefore its linear relaxation provides a better lower bound.

$$\sum_{\tau=t}^{T} x_{w\tau} + \sum_{\tau=0}^{min\{t+p_w-1,T\}} x_{w'\tau} \leq 1 \ \forall (w,w') \in \mathcal{A},\ t \in (0, T-1) \tag{3.11}$$

The main drawback of both discrete time formulations is the increase in the number of variables as the time horizon grows.

3.5.2 Continuous time formulation

In this formulations, the continuous variable S_w is used to represent the starting time of activity w. A set of binary variables $x_{ww'}$ is used for the precedence constraints, where $x_{ww'} = 1$ if activity w' cannot start until activty w is completed and 0 otherwise. This formulations are similar to the one of the most used for the Traveling Salesman Problem (TSP).

Forbidden Sets Formulation (FS)

Alvarez-Valdés and Tamarit ([AV93]) proposed a formulation based on the definition of forbidden Sets. A forbidden set is a subset of activities $F \subset W$ if between its activities there are no precedence constraints but they cannot be processed in parallel due to the resource constraints. That is:

- $x_{ww'} = 0 \ \forall (w,w') \in F$
- $\exists k \in R | \sum_{w \in F} b_{wk} \geq B_k$

A forbidden set F is a minimal forbidden set it neither of it subsets is a forbidden set itself ($\forall F' \subset F,\ \sum_{w \in F'} b_{wk} \leq B_k$). Feasible schedules are obtained by breaking all the minimal forbidden sets, that is, including a new precedence relation between at least two of its activities in order to comply with the resource constraints. Being $\mathcal{F}$ the set of minimal forbidden sets, the complete FS formulation is:

$$Minimize\ S_{n+1} \tag{3.12}$$

Subject to:

$$x_{ww'} + x_{w'w} \leq 1 \; \forall (w, w') \in W, w < w' \tag{3.13}$$

$$x_{ww'} + x_{w'w''} - x_{ww''} \leq 1 \; \forall (w, w', w'') \in W \tag{3.14}$$

$$x_{ww'} = 1 \; \forall (w, w') \in \mathcal{A} \tag{3.15}$$

$$S_{w'} - S_w - Mx_{ww'} \geq p_w - M \forall (w, w') \in W \tag{3.16}$$

$$\sum_{(w,w') \in F} x_{ww'} \geq 1 \; \forall F \in \mathcal{F} \tag{3.17}$$

$$x_{ww'} \in \{0, 1\} \tag{3.18}$$

$$S_w \geq 0 \; \forall w \in W \tag{3.19}$$

M is a big enough number, that can be set to un upper bound of the project makespan ($\sum_w p_w$). The objective function value is obtained directly by the starting time of the dummy final activity. Constraints 3.13 and 3.14 prevent the cycles on the sequencing of the activities. The precedence constraints are covered by 3.15. The link between variables $x_{ww'}$ and the activities starting time is made by 3.16. Finally, the minimal forbidden sets are broken by 3.17, as at least one sequencing decision must be taken per minimal forbidden set. Constraints 3.18 and 3.19 are domain restrictions.

This formulation has $O(n^2 + n)$ variables and $O(2^n|n^3)$ constraints. Due to the exponential growth of the number of constraints and the difficulties in indentifying the minimal forbidden sets, this formulation is barely used in practice.

Flow-Based Continuous Time Formulation (FCT)

Described by Artigues [AMR03], it shares with the previous formulation the use of the starting times S_w and sequencing $x_{ww'}$ variables. However, instead of forbidden sets, it uses a continuous variable $f_{ww'k} = (0, 1...B_k)$ to represent the quantinty of resource k that is transfered from activity w (at the end of its processing) to activity w' (at the start of its processing). It sets the dummy start activity as a resource source and the end activity as a sink ($b_{0k} = b_{n+1k} = B_k$).

$$Minimize \quad S_{n+1} \tag{3.20}$$

Subject to:

$$x_{ww'} + x_{w'w} \leq 1 \ \forall (w, w') \in W, w < w' \tag{3.21}$$

$$x_{ww'} + x_{w'w''} - x_{ww''} \leq 1 \ \forall (w, w', w'') \in W \tag{3.22}$$

$$x_{ww'} = 1 \ \forall (w, w') \in \mathcal{A} \tag{3.23}$$

$$S_{w'} - S_w - M x_{ww'} \geq p_w - M \forall (w, w') \in W \tag{3.24}$$

$$f_{ww'k} - min\{b_{wk}, b_{w'k}\} x_{ww'} \leq 0$$

$$\forall (w \in W - \{n+1\}; w' \in W - \{0\}, k \in R) \tag{3.25}$$

$$\sum_{w'=0}^{n+1} f_{ww'k} = b_{wk} \ \forall (w \in W, k \in B) \tag{3.26}$$

$$\sum_{w'=0}^{n+1} f_{w'wk} = b_{wk} \ \forall (w \in W, k \in B) \tag{3.27}$$

$$f_{n+1,0,k} = B_k \tag{3.28}$$

$$x_{ww'} \in \{0, 1\} \tag{3.29}$$

$$S_w \geq 0 \ \forall w \in W \tag{3.30}$$

$$f_{ww'k} \geq 0 \ \forall (w, w') \in W, \ k \in R \tag{3.31}$$

Constraints 3.21 to 3.24 remain the same as in the Forbidden Sets formulation. Only a positive flow is permited between tasks and this flow must be not greater than the resource consumption of the ending / starting task, as per 3.25. The input / output flows from a task must be both equal to the resource consumption: constraints 3.26 and 3.27 are the resource balance per activity. The total amount of a resource k on the flow is limited by 3.28. Finally, 3.29 to 3.31 are the domain constraints.

This formulation provides a poor relaxation, compared to discrete time formulations, although it can be preferable to them for instances involving large time scale. On top of this, it has been a widely studied formulation on its application for the TSP.

3.6 RCSP Event Based Formulations

Event Based Formulations for the RCSP where developed by Kone in 2009 [Kon09], from a model introduced by Zapata, [ZJ08]. These formulations define a series of events which correspond to the start or end of the different activities. They are based on the fact that for the RCSP it always exists an optimal semi-active schedule in which the start time of an activity is either 0 or coincides with the completion time of another activity [Spr95]. Therefore, at most $n + 1$ events have to be considered, and only n for the case of the second Event Based Formulation that we will discuss.

They have the advantage of not depending on the time horizon, making them especially relevant for long time projects. Koné proved that for some instances these formulations, and more precisely, the On/Off Event Based outperformed all the previous MILP formulations. On another set of instances these formulations even obtained better results than a specialised heuristic method.

3.6.1 Start/End Event Based Formulation (SEE)

The Start/End Event Based Formulation (SEE) involves two types of binary variables,x_{we} and y_{we}, that are equal to 1 if task w starts (in the case of x_{we}) or ends (y_{we}) at event e and 0 otherwise. The time of each event is represented by the continuous variable t_e. As for the resources, the variables r^*_{ke} represent the amount of resource k needed immediately after the event e. The set of events is represented by $\mathcal{E}$.

The SEE formulation can be written as follows:

$$Minimize \quad t_n \tag{3.32}$$

Subject to:

$$t_0 = 0 \tag{3.33}$$

$$t_{e+1} - t_e \geq 0 \ \forall e \in \mathcal{E} \backslash \{n\} \tag{3.34}$$

$$\sum_{e \in E} x_{we} = 1 \ \forall w \in W \tag{3.35}$$

$$\sum_{e \in E} y_{we} = 1 \ \forall w \in W \tag{3.36}$$

$$\sum_{f \in \mathcal{E}} f y_{wf} - \sum_{e \in \mathcal{E}} e x_{we} \geq 1 \ \forall w \in W \tag{3.37}$$

$$t_f - t_e - p_w x_{we} + p_w(1 - y_{wf}) \geq 0 \ \forall w \in W, (e, f) \in \mathcal{E}, e < f \tag{3.38}$$

$$\sum_{e'=e}^{n} y_{we'} + \sum_{e'=0}^{e-1} x_{w'e'} \leq 1 \ \forall (w, w') \in A, \forall e \in \mathcal{E} \tag{3.39}$$

$$r^*_{0k} = \sum_{w \in W} b_{wk} x_{w0} \ \forall k \in R \tag{3.40}$$

$$r^*_{ek} = r^*_{e-1k} + \sum_{w \in W} b_{wk} x_{we} - \sum_{w \in W} b_{wk} y_{wk} \ \forall e \geq 1, k \in R \tag{3.41}$$

$$r^*_{ek} \leq B_k \ \forall e \in \mathcal{E}, k \in R \tag{3.42}$$

$$x_{we} = \{0, 1\} \ \forall (w \in W, e \in \mathcal{E}) \tag{3.43}$$

$$y_{we} = \{0, 1\} \ \forall (w \in W, e \in \mathcal{E}) \tag{3.44}$$

$$t_e \geq 0 \ \forall e \in \mathcal{E} \tag{3.45}$$

The objective 3.32 is to minimize the time of the final event. The first event must start at the beginning of the project, 3.33, and events must be oredered in time, although two of them may have the same time, as per 3.48. Constraints 3.35 and 3.36 assure that each activity has one and only one start/end event. Constraint 3.37 means that the end event must be after the start event. The duration of each activity, that must be at least its processing time, is assured by 3.38. The resource precedence constraints are 3.39.

As for the resources, the total available resources per type are limited by 3.42. The resources needed inmediately after the first event are 3.40. From them onwards, the resources needed after each event are calculated by the difference between the once consumed by the activities that end at event e and the ones beginning at that event 3.41 .

To end with, 3.43 and 3.45 are the domain constraints.

This formulation involves $O(2n(n+1))$ binary variables, $(n+1)$ continuous variables, and $(n+1)(|R| + n^2/2 + |A|) + 3n$ non-redundant constraints. Compared with DT and DDT, this formulation involves a polynomial number of variables and constraints.

3.6.2 On/Off Event Based Formulation

On/Off Event Based Formulation (OOE) uses only one set of binary variables. This variables: z_{we} equal to 1 if activity w begins or is still active inmediately after event e. Therefore, only n events are needed. As for the SEE based formulation, the continuous variable t_e is used for the time at which event e takes place.

In this formulation, there is no need for resource variables, as they can be controlled on an easier way. However, a new continuous variable C_{max} measures the project completion time.

$$Minimize \qquad C_{max} \tag{3.46}$$

Subject to:

$$t_0 = 0 \tag{3.47}$$

$$t_{e+1} - t_e \geq 0 \; \forall e \in \mathcal{E} \backslash \{n-1\} \tag{3.48}$$

$$C_{max} \geq t_e + (z_{we} - z_{we-1})p_w \; \forall e \in \mathcal{E}, w \in W \tag{3.49}$$

$$t_f - t_e \geq ((z_{we} - z_{we-1}) - (z_{wf} - z_{wf-1} - 1)p_w \; \forall (e,f) \in \mathcal{E}, f > e, w \in W \tag{3.50}$$

$$\sum_{e'=0}^{e-1} z_{we'} \leq e(1 - (z_{we} - z_{we-1})) \; \forall e \in \mathcal{E} - \{0\} \tag{3.51}$$

$$\sum_{e'=e}^{n-1} z_{we'} \leq (n-e)(1 + (z_{we} - z_{we-1})) \; \forall e \in \mathcal{E} - \{0\} \tag{3.52}$$

$$z_{we} + \sum_{e'=0}^{e} z_{w'e'} \leq 1 + (1 - z_{we})e \; \forall e \in \mathcal{E}, (w,w') \in A \tag{3.53}$$

$$\sum_{e=0}^{n-1} b_{wk} z_{we} \leq B_k \; \forall e \in \mathcal{E}, k \in R \tag{3.54}$$

$$z_{we} = \{0,1\} \; \forall w \in W, e \in \mathcal{E} \tag{3.55}$$

$$t_e \geq 0 \; \forall e \in \mathcal{E} \tag{3.56}$$

Most of this formulation is based on the fact that $(z_{we} - z_{we-1}) = 0$ for most of the events, with the exception of the event after which an activity starts/ends, when $z_{we} - z_{we-1} = 1$ for the start and $z_{we} - z_{we-1} = -1$ for the end.

The objective function is represented by C_{max}, calculated in 3.49 as the sum of the last event in which a task starts and its completion time. The order of the events (3.48) and the starting of the project (3.47) is similar to the SEE formulation.

For each activity, 3.51 and 3.52 are non-preemption constraints: if activity w starts at event e it cannot be active before this event, and if it ends at event e it cannot be active afterwards. A minimum duration for the activity equal to its processing time is required by 3.50.

Constraints 3.53 are the precedence constraints and 3.54 the resource constraints. Variable domains are expressed by 3.55 and 3.56.

This formulation involves $O(n^2)$ binary variables (twice as few as SEE and the same number as FCT), $(n+1)$ continuous variables, and $(n+1)(3+|A|+|R|+n^2)+n^2n$ non-redundant constraints.

3.7 RCSP Instance Classification Indicators

On the previous section, several solving methods have been presented. It has already been said that some of them may have significantly different performances depending on the data. In order to help distinguish the different possible data sets for a RCSP, Artigues [AKL$^+$10], provides a complete study of instance classification and indicators, as well as a benchmark of different well known RCSP instance sets.

The RCSP has a complex structure and can be decomposed into many categories depending on the resource features, the activity requirements and the precedence graph. According to this classification options, there are standard instance indicators that focus on:

- Precedences
- Resources
- Time
- Hybrid
- Upper bound based indicators

This indicators have been developed and tested for the most standard RCSP with precedence constraints and whose objective is the minimization of the Project Lead Time, [AKL$^+$10]. However, they are of interest even for other instances in order to guide on the obtention for more adequate indicators for each case.

3.7.1 Precedence-Oriented Indicators

They evaluate the complexity of the resource network. This kind of measures have been introduced since 1966, when Pascoe first defined the Network Complexity Measure. Together with it, the Order Strength and the Complexity Index have been the most common indicators refered to in the litterature, as reviewed by Demeulemeester [EDW03]. For them, it is important to calculate the Transitive Closure of the precedence graph. This is, to find all pairs of vertices in the input graphs that are connected via non null paths.

OS: Order Strength

Studied by Herroelen, [HR99]. It is defined as the number of precedence relations, including the transitive ones, divided by the theoretical maximum of such precedence relations, namely $w(w-1)/2$, where w denotes the number of activities. It can also be defined as the density of the transitive closure of the precedence graph.

NC: Network Complexity

The network complexity is the average number of arcs per task, w, when all the redundant constraints have been eliminated from the graph. Although Kolish et al, [RKD95] referred to this factor as marginally significant, Elmaghraby and Heroellen [EH80] observed that instances with similar NC may have different precedence graph structures, implying different solution hardness.

CI: Complexity Index

It is used to determine how nearly series parallel the precedence graph is, defined as the minimum number of node reductions required to reduce the precedence graph into a sigle arc graph.

Progressive level PL_i and regressive level RL_i indicarors

More recently, Tavares and Vanhoucke ([TFC99], [MV04]) developped a new set of indicators that rely on the progressive level PL_i and regressive level RL_i for each activity $i \in \mathcal{V}$. This concepts are defined as follows:

$$PL_i = max_{j \in P_i}(PL_j) + 1 \quad \forall P_i \neq \emptyset$$

$$RL_i = min_{j \in SU_i}(RL_j) - 1 \ \ \forall SU_i \neq \emptyset$$

The maximal progressive level is denoted by m. If $P_i = \emptyset$ then $PL_i = 1$ by definition, whereas if $SU_i = \emptyset$ then $RL_i = m$.

The Tavares and Vanhoucke ([TFC99], [MV04]) indicators, based on those two concepts are:

I_1**: Size of the Problem** It measures the total number of activities $n = card(\mathcal{W})$

I_2**: Serial or Parallel Indicator** As the CI, it measures how nearly series parallel the precedence graph is, but based on the progressive level m. It compares the progressive level of the graph with the progressive level of a total series graph, $n-1$:

$$I_2 = \frac{m-1}{n-1} \text{ if } n > 1; \text{ 1 otherwise}$$

I_3**: Activity Distribution** For each level ($a = 1...m$) we can define its width, w_a as the number of activities at that progressive level. Depending on the maximal width and its distribution and several shapes can be adopted to describe the function $W(a)$. The indicator measures the distribution of the activities over the progressive levels by the relationship between the total absolute deviations α_w and α_{max} where:

$$I_3 = \frac{\alpha_w}{\alpha_{max}} = \frac{\sum_{\alpha=1}^{m} |w_a - \overline{w}|}{2(m-1)(\overline{w}-1)} \ \text{ if } m \notin \{1, n\}; \text{ 0 otherwise}$$

α_{max} determines the maximal posible value for a graph with n activities and m progressive levels.

I_4 **and** I_5**: Short and Long Arcs** They measure the total number of short and long arces compared to the maximum posible.

I_6**: Topological Float** The float associated to each activity w measures the degree of freedom concerning the allocation of w to one of the scheduling stages. It is calculated as the difference between its regressive and progressive level. The indicator based on this is:

$$I_6 = \frac{\sum_{w \in \mathcal{W}} RL_w - PL_w}{(m-1)(n-1)} \quad \text{if } m \notin \{1, n\};\ 0 \text{ otherwise}$$

3.7.2 Resource Features Oriented Indicators

There are two main indicators that focus on the resource distribution and availability

RF: Resource Factor

The resource factor is the average number of required resources per activity divided per the total number of existing resources. It does not take into account the number of resources of a type that are needed but only the type of resources needed per task:

$$RF = \frac{\sum_{w \in \mathcal{W}} z_{wk}}{n \|\mathcal{R}\|}$$

Although Kolish et al, [RKD95], showed that the instance harness increases as RF increasesit is important to remark, that it can be strongly biased if there is a significant number of activities without any resource requirement.

RC: Resource Constrainedness

This indicator compares the level of resource R_k availability compared to the average level of activity requirements for each resource $R_k \in \mathcal{R}$:

$$RC_k = \sum_{w \in \mathcal{W}} b_{wk} / n B_k = \overline{b_{wk}} / B_k$$

Its impact on the instance resolution follows an easy-hard-easy bell curve, as the extreme points are those where there is no effective resource constraint or little variation is allowed due to the resource tight constraints.

3.7.3 Time Features Oriented Indicators

FFR: Free Float Ratio

The free float ratio is measure of the flexibility of the start time of a task without afecting the project completion time:

$$FFR = \frac{\sum_{w \in \mathcal{W}} p_w}{\sum_{w \in \mathcal{W}} (LS_w - ES_w + p_w)}$$

This ratio can take values $0 < FFR < 1$. As the FFR decreases, the tasks have less flexibility and therefore the instances become harder.

3.7.4 Hybrid Indicators

This indicators include two or more of the above features and therefore provide a more global instance classification.

RS: Resource Strength

Incorporates both resource and time features, as it expresses the relationship between the resource demand of the jobs and the resource availability. It was defined in order to overcome the disadvantages of the resource constrainedness, as this indicator is not reliable in terms of feasibility or comparatios, as it does not take into account the distribution of the resource demand allover the project.

In order to cope with that drawback, we define a minimal and maximal resource demand for the project and let the resource availability be a combination of both, scaled by the Resource Strength:

$$B_k = B_k^{min} + RS(B_k^{max} - B_k^{min})$$
$$RS = \frac{(B_k - B_k^{min})}{(B_k^{max} - B_k^{min})}$$

Where B_k^{min} is the maximal of the demands of resource k per task, taking into account the less demanding mode per task and B_k^{max} is the peak demand for the earliest precedence feasible schedule.

However, the reliability of this indicator is still moderate, as once a task demands the whole resource availability ($B_k = B_k^{min}$), $RS = 0$ independently from the rest of the demands.

DR: Disjunction Ratio

The disjunction ratio integrates resource and precedence features:

$$DR = |TE \cup D|/(w(w+1)/2)$$

Where TE is the transitive closure (used also for the OS indicator) and D is the number of pairs of activities in disjunction, this is, the pairs of activities that cannot be performed in parallel due to resource constraints. Baptiste et le Pape [BP00] defined this indicator to distinguish between cumulative instances (low disjunction ratio) and disjunctive instances (high disjunction ratio), as they may be tuckled by different king of techniques.

FS and AFS:Number and Cardinality of Minimal Forbidden Sets

As it has been said on section 3.5.2, in order to solve a RCSP, each minimal forbidden must be broken, decision which implies $|F||F-1|$ options. The FS and AFS indicators measure the number and cardinality of the minimal forbidden sets.

Stork [SU05] showed that this indicators increase with the RF and with them the instance hardness. Artigues et al [AKL$^+$10] proposed an improvement of this indicators, taking into account only the minimal forbidden sets violated by the earliest precedence feasible schedule.

3.7.5 Indicator Summary

Table 3.2 provides a summary of the main indicators we have mentioned, their hardness, relevance and the main authors who introduced them. The main references on this issue are [HR99], [RKD95], [BKFS92], [AKL$^+$10], [BP00], [SU05]

Throughout the literature, several indicators have been proposed for measuring an instance hardness. Up to now there is not a clear dominance rule between those indicators. Although some of them, on their own have been prooved as reliable, there is not a study about the impact of different indicator combinations. Moreover, their predictive power depends not only on the instance but also on the solution procedure to be used.

Table 3.2: Main RCSP Instance Indicators

RELATED TO	ABR.	INDICATOR	HARDNESS	RELEVANCE	MAIN CITE
Precedence Graph	OS	Order Stregth	High (Transitive Closure)	High	[HR99]
Precedence Graph	NC	Network Complexity	Easy	Not clear	[RKD95]
Precedence Graph	CI	Complexity Index	Algorithm needed	Low	[BKFS92]
Resource Features	RF	Resource Factor s	Easy	Seriously biased	[RKD95]
Resource Features	RC	Resource Constrainedness	Easy	Easily Biased	
Time Features	FFR	Free Float Ratio	Medium		
Hybrid	RS	Resource Strength	Medium	Moderately	[RKD95]
Hybrid	DR	Disjunction Ratio	High(Transitive Closure)	High	[BP00]
Hybrid	FS	Nb. of Minimal Forbidden Sets	Algorithm	Low	[SU05]
Hybrid	ACFS	Av. Cardinality of Minimal FS	Algorithm	Low	[SU05]
Hybrid	UFS	Unsolved Minimal FS			
Hybrid	UACFS	Average Cardinality of UFS			

3.8 Conclusion and Master Thesis Scope

On the one hand, within the industrial context review from chapter 2 it has been identified a need of scheduling methods suitable for the aeronautical industry, more precisely, for the scheduling of works within an assembly platform.

On the other hand, in this chapter, we have presented the structure and characteristics of a resource constraint scheduling problem (RCSP). Also, we have discussed the existance of the time constrained scheduling problem, addressed in the literature as an especific case of the former. Moreover, we have introduced important definitions, as the notions of different types of schedules (active, semi-active, left-shifted). A review of the existing solution methods for RCSP has been provided, focusing on MILP formulations. We have finally gone through of the main instance indicators, their hardness, relevance and the main authors who introduced them.

Currently, the most common available scheduling methods on the industry are heuristic approaches. However, the recent developments on exact optimization methods and metaheuristics, together with computational capacity, justify the search of a linear programming formulation for our problem. At the same time, although a wide range of MILP formulations have been addressed in the litterature none of them are suitable for our specific problem. Moreover, the TCSP has been rarely refered to but it is the approach needed for all industries with takt paced production.

Secondly, throughout the litterature review we have stated the importance of the characterization of the instances, as each set of data is solved in a more effiecient way by different models. The existint indicators are not enough for the characterization of aeronautical platforms' data sets. Therefore, a new set of indicators is needed to distinguish among different data sets that may be otherwise confused into a single case.

Taking the previous needs into account, in order to cover the knowledge gap identified for TCSP, the scope of this work has been twofold: to **develop of a valid linear programming formulation for the aeronautical platform scheduling problem**, to provide guidance for an efficient implementation and also to address the need of a **new set of indicators** suitable for both the aeronautical platform scheduling problem and its formulation.

Chapter 4
The APSP: Aeronautical Platform Scheduling Problem

As it has been explained on chapter 2, aeronautical assembly lines consist on a series of platforms where different works are executed. Each product has to go through all the platforms. At the same time, the line is syncronised, this means that the time that each product remains on a platform is always the same and equal to the rate at which the assembly line produces its output.

Usually, both final and intermediate assembly lines are dedicated to only one product family. However, there may be some specific installations between the products of a same family. In that case, an effort is made in order to guarantee that the takt time remains standard throughout the line. Different strategies for that, refered to the aeronautical industry, can be found in [HRS+01].

The assignment of the works to a platform is in most of the cases related to industrial issues: assembly technologies and the need of specific jigs that cannot be easily moved. Therefore, the asssignment of works between platforms is made only once, during the line definition. Although there may be a small percentage of works that can be done in more than one platform, once the works are assigned they are rarely moved from one platform to another, so we can assume with no loss of generality that the task assignment is constant per platform.

Taking this into account, the scheduling decision consists on **establishing the order in which the works will be done together with the resources allocated to each of them, given the line takt time and a set of works per platform** .

On the recent years, automatisation of aeronautic platforms has experienced major improvements. In spite of this, aeronautical assembly remains intensive in highly-qualified operators. Oftenly, some of the works need to be done by workers with a specific certification, and not all them have the same certifications. These different skills are managed by the use of 'profiles' that gather one or more certifications.

Each operator is assigned a profile, according to his skills. As well as this, each task may be done by operators with one or several profiles. For example, if *profile 1* includes only elementary mechanical tasks (such as drilling and riveting) and *profile 2* includes the previous ones and also pipes installation, a work involving riveting and drilling can be done by both profiles.

Operators are organized in permanent groups, with a team leader per group of 5 to 15 operators. Each group is responsible of tracking its performance and taking corrective actions. In order to enable this, each group works together and is assigned to a platform for the complete takt time. Extra resources may be available for peak demands, but they must not be included on the stardard schedule. As a result, if N operators are needed only one day at one platform, they will stay on the platform within the takt time, and the related cost must be allocated to that platform.

Some of the assembly works can be performed by only one operator, whereas others must be done by two or more. On some cases, there is a range of possible operator numbers, which will lead to a set of production times for the work. As well as this, having more than the minimum number of operators reduces the lead time, but not in a linear way, as some activities may not be performed in parallel. For example: the lead time for preparing the necessary material and reading the documentation will be the same no matter what is the number of operators working. As a result, the number of operators reduces the lead time in a non-linear manner specific per work.When several operators are involved, we will assume that they all need to have the same profile, as that is how they work in practice.

Platforms can be of very different sizes: from a part of a major component (a fan cowl, for example) of 1-10 meters, to a complete aircraft (hundreds of meters) in the last steps of a final assembly line. However, even the smallest platforms are big enough for two or more jobs to be performed in parallel. Usually, several tasks are to be done near to each other, in a way that they cannot be executed at the same time due to space restrictions. As a result, platforms are divided on smaller working areas, where a limited number of operators can work at a time. This transforms the space on each area in a scarce renewable resource for scheduling.

Each task may require a set of tools that can be standard or specific for the work. If they are specific, different works rarely share a tool and if they share it, the tool can be duplicated if works are to be performed in parallel. Standard tools are available so that all operators can use them freely. Therefore, tools need not be included in the scheduling model.

The constraints between tasks can be of different nature. The most common case is that of precedence constraints: when a task cannot be started until a previous one has been finished. For example: harnesses cannot be routed until the supports to which they are attached have been riveted to the aircraft structure, or a funcional test cannot be performed until the system it tests has been completely installed.

Another kind are non-parallel constraints: these mean that some activities can be done in any order as far as they are not being performed at the same time. This is the case of some tasks that due to health and security reasons must be done with as fewer persons as possible in the hangar, e.g. corrosion inhibition application. This also happens in tests that require a specific aircraft condition: hydraulic tests need to have the power on, but the aircraft must have its power off for fuel tests.

Finally, maximum time lags also occur. For example, after sealing some part of the aircraft, the sealant must cure and therefore no work must be done near it until some time after the end of the task. Bonding tests have to be performed at the end of the installations, and bonding protection must be accomplished within the same working day where the test was passed. In some cases, not only a maximal time lag but also the same working team is required.

In accordance to the fixed takt time, the objective function will be to minimise the resource consumption per platform. As we have said, this is equivalent to minimising the sum of the peak operators demand per profile.

Taking all these into account and according to the general notation on section 3.2, the Aeronautical Platform Scheduling Problem is classified as :

$$MPS_m, \sigma,\ \rho \mid prec, temp, LT \mid \sum c_o * num_O$$

- $\alpha = MPS_m, \sigma, \rho$. This stands for a multimode resource constraint project where each activity can be processed in several alternative modes and exists a set of renewable resources available for each time period during the project execution: m being the resources, σ the units of each resource available and ρ the maximum number of units of the resources demanded by an activity. For our particular problem, the activities re the work tasks assigned to each platform. The renewable resources are the number of operators (each of them belonging to a particular skill) and the space in each of the platform's working areas. As well as this, each mode for an activity defines a combination of operator skills, number of operators and durations. All the operators assigned to an activity must be from the same skill and the range of possible numbers of allocated operators per tasks is independent from the chosen skill.
- $\beta = temp$. The set of temporal constraints includes different kinds. There are precedence constraints (task w' can not start until task w has been completed), non-parallel constraints (tasks w and w' cannot be in progress at the same time, but there is no precedence relation between them), and maximal time

lags between tasks (task w' must start within a maximal time after w has been completed). All the temporal constraints are independent from the mode in which a task is executed. As for maximal time lags, the specific case of tasks that must be done by the same team and with a maximum delay between them will be treated by pre-processing the data and transforming them into a single work, and therefore they are not included in the model.

- $\gamma = \sum c_o * num_O$ The objective function is to minimize the resource investment. The total lead time is fixed by the assembly line Takt Time. In consequence, the objective function is to minimise the labor cost of the assembly. As the operators once assigned to a platform stay working on it for all the Takt Time, minimising the labor cost is equivalent to minimising the maximum number of operators needed throughout the Takt Time.

On average, for final assembly lines, the takt time varies from 14 to 25 working days. There are two or three operator teams per platform, that is, more or less 20 operators in shifts of 8 hours. The most common profiles are: mechanical installator, electrical installator, fluid systems installator, inspector and test specialist. However, as works within a platform are usually from similar technologies no more than three profiles are required per platform. As for the working areas, there can be defined up to 5 different ones per platform. The total workload can be from 1000 to 3000 manhours.

Precedence constraints are the most frequent. There are normally a small percentage of works that do not have any kind of precedence relationship with the others. Works are oftenly organised on severall groups (1-10) that can be done in parallel. Within this groups, most of the task must be done in series. Non-parallel constraints and maximal time lags occur at a much lower rate than precedence constraints but, at the same time, they can not be relaxed because they have a major impact on product quality, reliability or health and safety issues.

Chapter 5

The APSP: Multimode Event Based Formulation

From the different formulations, refered to on section 3 , we will use on this section both of the Event Based Formulations. The Start/End Event (SEE) and ON/OFF (OOE) formulations will be addressed and improved in order to be able to use them for our specific problem.

5.1 Sets, Parameters and Variables

We will use four Sets, common to both SEE and OOE extended formulations, they are listed on table 5.1. The parameters needed to define our model are listed on table 5.2 and the common variables for both Evet Based Formulations are listed on table 5.3.

5.2 Start End Event Based Multimode Formulation: SEE-M

The Start-End Event formulation proposed by [KALM11] and refered to on section **??** deals with a single mode RCSP. For our problem, we need to deal with a multi mode case, as each task may be performed by different operator profiles o and number of operators n. To deal with this, the variables x_{we} and y_{we} have been replaced by a sets of variables x_{weop} and y_{weop}, where $x_{weop} = 1$ ($y_{weop} = 1$) if task w starts (ends) at event e using p operators of profile o. Additional constraints are added to

Table 5.1: Sets

$\mathcal{O}$	:	All the operator Profiles. Index: o. Card $(\mathcal{O}) = O$
$\mathcal{W}$	:	All the Work Tasks. Index: w. Card $(\mathcal{W}) = W$
$\mathcal{A}$	:	All the Working Areas. Index: a. Card $(\mathcal{A}) = A$
$\mathcal{E}$	:	Set of events. The number of events depends on the formulation and it is related to the number of tasks to be scheduled. Index: e, e', e'', f .

Table 5.2: Parameters

D_w	:	Total amount of working hours for task $w \in \mathcal{W}$, if assigned only to one operator; $\forall w \in \mathcal{W}$
Γ_{pw}	:	Reduction coefficient to obtain the work task's w's makespan when it is done by n operators; $\forall w \in \mathcal{W}$, $MIN_w^{op} \leq p \leq MAX_w^{op}$
P_{ow}	:	1 if task $w \in \mathcal{W}$ can be done by operators with profile o and 0 otherwise; $\forall w \in \mathcal{W}$, $o \in \mathcal{O}$
MAX_w^{op}	:	Maximum number of operators that can work on task w; $\forall w \in \mathcal{W}$
MIN_w^{op}	:	Minumum number of operators that can work on task w; $\forall w \in \mathcal{W}$
$PRE_{ww'}$	:	1 if the precedence graph includes a precedence relationship between work tasks w, w' and 0 otherwise; $(w, w') \in \mathcal{W}$
$NONP_{ww'}$	:	1 if the precedence graph includes a non-parallel constraint between work tasks, w, w' and 0 otherwise; $(w, w') \in \mathcal{W}/w < w'$
$MTL_{ww'}$	:	1 if work task w' is to be begun inmediately after w; $(w, w') \in \mathcal{W}$. $(MTL_{ww'} \leq PRE_{ww'})$.
Δ	:	Maximum time between the end of task w and the beginning of task w' when there exists a maximal time lag constraint between them.
$AREA_{aw}$	:	1 If work task w is done on area a, 0 otherwise; $\forall a \in \mathcal{A}$, $w \in \mathcal{W}$
CAP_a	:	Maximum number of operators that can work in area a; $\forall a \in \mathcal{A}$
TT	:	Platform Takt time
M	:	Large enough number. This can be fixed as $M = max(TT, E)$

Table 5.3: Common Event Formulation Variables

t_e	:	Time of event e
num_o^{op}	:	total number of operators of profile o needed, $o \in \mathcal{O}$
t_w^i	:	Defines the starting time of task w. This will be used for maximal time lag constraints, and therefore defined $\forall w \in \mathcal{W}$ and $\sum_{w'} MTL_{ww'} + MTL_{w'w} > 0$

guarantee the same asignation of operator profiles and number throughout the task fullfilment.

We define the non-negative variables r^*_{oe} to represent the amount of resource o needed immediately after event e and non-negative variables s^*_{ae} to represent the number of operators working on area a immediately after event e.

Finally, on top of the precedences constraints from the original model, others dealing with non-parallel tasks and maximal time lags have been included.

5.2.1 Variables for Start-End Event Based multimode formulation

x_{weop}	:	1 if work task w starts at event e with p operators of profile o and 0 otherwise; $\forall w \in \mathcal{W}, e <> last(\mathcal{E}), MIN^{op}_w \leq p \leq MAX^{op}_w, \forall o/P_{ow} = 1$
y_{weop}	:	1 if work task w ends at event e with p operators of profile o and 0 otherwise; $\forall w \in \mathcal{W}, e <> first(\mathcal{E}), MIN^{op}_w \leq p \leq MAX^{op}_w, \forall o/P_{ow} = 1$
r^*_{oe}	:	Number of operators of type o required by the tasks in progress immediately after event e, $\forall o \in \mathcal{O}, e \in \mathcal{E}$
s^*_{ae}	:	Number of operators on zone a required by the tasks in progress immediately after event e, $\forall a \in \mathcal{A}, e \in \mathcal{E}$
$\alpha_{ww'}$	:	1 if w ends before w' starts and 0 vice-versa. Defined $\forall w, w'/NONP_{ww'} = 1$

For this formulation, the number of events is equal to the number of work tasks plus 1: $E = W + 1$.

5.2.2 SEE-M Formulation

$$Minimize \sum_o num^{op}_o \tag{5.1}$$

$$Minimize \sum_o num^{op}_o LT - \sum_w DUR_w \tag{5.2}$$

Subject to:

$$t_0 = 0 \tag{5.3}$$

$$t_e \leq TT \ \forall e \neq \{0\} \tag{5.4}$$

$$t_{e+1} - t_e \geq 0 \ \forall e \neq last(e) \tag{5.5}$$

$$\sum_{eop} e y_{weop} - \sum_{eop} e x_{weop} \geq 1 \quad \forall w \in \mathcal{W} \tag{5.6}$$

$$\sum_{eop} x_{weop} = 1 \ \forall w \in \mathcal{W} \tag{5.7}$$

$$\sum_{eop} y_{weop} = 1 \ \forall w \in \mathcal{W} \tag{5.8}$$

$$t_f - t_e - \sum_{o} D_w \Gamma_{pw} x_{weop} + (D_w \Gamma_{pw})(1 - \sum_{o} y_{wfop}) \geq 0$$
$$\forall (f, e) \in \mathcal{E}, f > e, w \in \mathcal{W}, MIN_w^{op} \leq p \leq MAX_w^{op} \tag{5.9}$$

$$\sum_{eo} x_{weop} = \sum_{eo} y_{weop} \ \forall w \in \mathcal{W}, MIN_w^{op} \leq p \leq MAX_w^{op} \tag{5.10}$$

$$\sum_{ep} x_{weop} = \sum_{ep} y_{weop} \ \forall w \in \mathcal{W}, o \in \mathcal{O} \tag{5.11}$$

$$\sum_{\substack{e''=0 \\ o,p}}^{e-1} x_{w'eop} + \sum_{\substack{e'=e \\ o,p}}^{E} y_{we'op} \leq 1 \ \forall e \in \mathcal{E}, (w, w') \in \mathcal{W}/PRE_{ww'} = 1 \tag{5.12}$$

$$t_w^i \geq t_e - M(1 - \sum_{op} x_{weop}) \ \forall w \in \mathcal{W} \ / \ \sum_{w'} (MTL_{ww'} + MTL_{w'w}) > 0 \tag{5.13}$$

$$t_w^i \leq t_e + M(1 - \sum_{op} x_{weop}) \ \forall w \in \mathcal{W} \ / \ \sum_{w'} (MTL_{ww'} + MTL_{w'w}) > 0 \tag{5.14}$$

$$t_{w'}^i - t_w^i - \sum_{eop} \Gamma_{wp} D_w x_{weop} \leq \Delta \quad \forall (w, w') \in \mathcal{W} \ / \ MTL_{ww'} = 1 \tag{5.15}$$

$$\sum_{eop} e y_{weop} - \sum_{eop} e x_{w'eop} \leq M(1 - \alpha_{ww'}) \quad \forall (w, w') \in \mathcal{W}/NONP_{ww'} = 1 \tag{5.16}$$

$$\sum_{eop} e y_{w'eop} - \sum_{eop} e x_{weop} \leq M(\alpha_{ww'}) \quad \forall (w, w') \in \mathcal{W}/NONP_{ww'} = 1 \tag{5.17}$$

$$r^*_{o0} - \sum_{wp} px_{w0op} = 0 \ \forall o \in \mathcal{O} \tag{5.18}$$

$$r^*_{oe} - r^*_{oe-1} + \sum_{\substack{w \\ P_{ow}=1}} (\sum_p py_{weop} - \sum_p px_{weop}) = 0 \ \forall o \in \mathcal{O}, e \in \mathcal{E} - \{0\} \tag{5.19}$$

$$r^*_{oe} \leq num^{op}_o \ \forall o \in \mathcal{O}, e \in \mathcal{E} \tag{5.20}$$

$$s^*_{a0} - \sum_{wop} px_{w0op} AREA_{aw} = 0 \ \forall a \in \mathcal{A} \tag{5.21}$$

$$s^*_{ae} - s^*_{ae-1} + \sum_w (\sum_{op} py_{weop} AREA_{aw} - \sum_{op} px_{weop} AREA_{aw}) = 0 \ \forall a \in \mathcal{A}, e \in \mathcal{E} - \{0\} \tag{5.22}$$

$$s^*_{ae} \leq CAP_a \ \forall a \in \mathcal{A}, e \in \mathcal{E} \tag{5.23}$$

$$x_{ewop} \in \{0,1\} \ \ \forall e \in \mathcal{E}/\{0\}, \ w \in \mathcal{W}, \ o \in \mathcal{O}/P_{ow} = 1, MIN^{op}_w \leq p \leq MAX^{op}_w \tag{5.24}$$

$$y_{ewop} \in \{0,1\} \ \forall e \in \mathcal{E}/\{last(\mathcal{E})\}, \ w \in \mathcal{W}, \ o \in \mathcal{O}/P_{ow} = 1, MIN^{op}_w \leq p \leq MAX^{op}_w \tag{5.25}$$

$$r^*_{oe} \geq 0 \ \forall e \in \mathcal{E}, \ o \in \mathcal{O} \tag{5.26}$$

$$0 \leq s^*_{ae} \leq CAP_a \ \forall e \in \mathcal{E}, a \in \mathcal{A} \tag{5.27}$$

$$t_e \geq 0 \ \forall e \in \mathcal{E}, (w, w') \in \mathcal{W} \ /\alpha_{w,w'} \in \{0,1\} \tag{5.28}$$

Objective Function The objective function is to minimize the total project cost. This cost depends only on the labor costs as this is the only one in which different solutions have an impact. The labor costs are propotional to the maximum number of operators needee during the planning horizon, as resources allocated to a work station will remain in it throughout the complete takt time. Function 5.1 deals with the labor costs in terms of the total amount of operators. Function 5.2 minimizes the non-saturation of ressources. Both of them would lead to the same resources.

Order of the events Constraint 5.3 forces the first event to begin at t=0 and constraint 5.4 assures that there is no delay in the station completion. The order of the events on time is imposed by constraint 5.5.

Start-End events for each task Constraint 5.6 states that the start event of a task must precede its end event. Constraints 5.7 and 5.8 limit to one the start and end per work task. Constraint 5.6 can be omitted. However, it defines a smaller polytope and therefore a better linear relaxation.

Time per task Constraint 5.9 fixes the minimum time difference between the start and the end events to the duration of the task.

One mode per task A single mode for performing the task is imposed by constraints 5.10 and 5.11. Note that the possible modes are limited by the definition of x_{weop} and y_{weop} as they are only defined for $MIN_w^{op} \leq p \leq MAX_w^{op}$ and $\forall o/P_{ow} = 1$.

Temporal constraints As for the relations between tasks: 5.12 is the multimode expression for the precedence constraints. It states that if w must precede w' then, if w finishes at event e, w' cannot start until event $e + 1$. Maximal time lags are imposed on constraints 5.13 to 5.15. Constraints 5.13 and 5.14 define the start time of a task. This constraints will only be calculated for the task involved on maximal time lag constraints, this is $\forall w \in \mathcal{W} \; / \; \sum_{w'} (MTL_{ww'} + MTL_{w'w}) > 0$. Given 5.13 and 5.14 the tasks, 5.15 limits the time between the end of a task and the start of its succesor. This time lag is defined as a constant parameter Δ but it can be easily replaced by a task dependent parameter: $\Delta_{ww'}$. In this constraints, M can be replaced by the platform's Takt time, TT.

Constraints 5.16 and 5.17 define the non-parallel constraints. In this constraints, M can be replaced by the number of events.

Ressource Needs The two sets of resources taken into accout are the operators and the space on each of the platform working areas. For operators, 5.18 is the number of operators needed per profile inmedately after the first event. Constraint 5.19 deals with the number of operators per profile, o, for all the other events. Constraint 5.20 calculates the maximum need of operators per profile allover the events. Similarly, constraints 5.21 and 5.22 assure that the maximum occupation in the platform working areas is not exceeded. Constraint 5.23 fixes the maximum capacity per area. This value can also be included as a domain restriction for s_{ae}^*, 5.27.

Domain Restrictions are equations 5.24 to 5.28 .

SEE-M Dimensions This formulation involves:

- $2EW \sum_w ((MAX_w^{op} - MIN_w^{op}) \sum_o P_{ow})$ start/end binary variables
- $|NONP|$ extra binary variables for the Non-parallel relations
- $OE + AE + E + MTL_w + O = E(O + A + 1) + MTL_w + O$ continuous variables, corresponding to $r_{oe}^*, s_{ae}^*, t_e, t_w^i$ and num_o^{op}

The number of constraints is:

$3W-1+E(\frac{(E-1)W}{2} \sum_w (MAX_w^{op} - MIN_w^{op})+2O+A+2)+\sum_w (MAX_w^{op} - MIN_w^{op})+$

$\sum_{wo} P_{ow} + |PRE| + |MTL| + 2MTL_w + 2|NONP| - O.$

Where: |PRE| $= \sum_{ww'} PRE_{ww'}$,|NONP| $= \sum_{w,w'} NONP_{w,w'}$,

|MTL|$= \sum_{ww'} MTL_{ww'}$ and MTL_w is the number of work tasks for which

$\sum_{w'} MTL_{ww'} + MTL_{w'w} > 0.$

For more detailed information about the number of constraints on SEE-M formulation, see appendix A.

At most $E = W + 1$. However, we will see that in some cases this number can be reduced. In that sense, it is important to note that the number of constraints is of the order of $(W + WE^2)$. This means that a reduction in the number of events has greater impact than lowering the number of work tasks.

In terms of W $(E = W + 1)$, the number of constraints for SEE-M formulation is:

$1+O+A+W(5+2O+A)+W^2(\frac{\sum_w (MAX_w^{op}-MIN_w^{op})}{2})+W^3(\frac{\sum_w (MAX_w^{op}-MIN_w^{op})}{2})+$

$\sum_w (MAX_w^{op} - MIN_w^{op}) + \sum_{ow} P_{ow} + |PRE| + |MTL| + 2MTL_w + 2|NONP|$

5.3 On-Off Event Based Multimode Formulation: OOE-M

In order to improve the formulation presented by Kone [KALM11] and refered to on section 3.6, we introduce two new sub-index for the binary variables z_{we}, replacing them by z_{weop}, equal to 1 if work task w is active at event e with p operators of profile o. Additional constraints are added to guarantee the same asignation of operator profiles and number throughout the task fullfilment.

In this formulation, there is no need for resource variables, as they can be controlled on an easier way.

Also, together with the precedence constraints from the original model, others dealing with generalized temporal constraints have been added.

5.3.1 Variables for On-Off Event Based multimode formulation

In addition to the variables defined in table 5.3, some OOE-M variables must be used:

z_{weop}	:	1 if work task w is active from event e to event $e+1$ with p operators of profile o , 0 otherwise; $\forall e \in \mathcal{E}, w \in \mathcal{W}, o \in \mathcal{O}/P_{ow} = 1, MIN_w^{op} \leq p \leq MAX_w^{op}$
β_{wop}	:	1 if work task w is performed by p operators of profile o, 0 otherwise; $\forall w \in \mathcal{W}, o \in \mathcal{O}/P_{ow} = 1, MIN_w^{op} \leq p \leq MAX_w^{op}$

5.3.2 OOE-M Formulation

$$Minimize \sum_o num_o^{op} \tag{5.29}$$

$$Minimize \sum_o num_o^{op} LT - \sum_w DUR_w \tag{5.30}$$

Subject to:

$$t_0 = 0 \tag{5.31}$$

$$\sum_{wop} z_{w0op} \geq 1 \tag{5.32}$$

$$t_{e+1} - t_e \geq 0 \ \forall e \neq \text{first}\mathcal{E} \tag{5.33}$$

$$\sum_{eop} z_{weop} \geq 1 \ \forall w \in \mathcal{W} \tag{5.34}$$

$$\sum_{\substack{o \\ MIN_w^{op} \leq p \leq MAX_w^{op} o}} \beta_{wop} = 1 \ \forall w \in \mathcal{W} \tag{5.35}$$

$$z_{weop} \leq \beta_{wop} \forall w \in \mathcal{W}, o \in \mathcal{O}, e \in \mathcal{E}, MIN_w^{op} \leq p \leq MAX_w^{op} \tag{5.36}$$

$$\sum_{e' \leq e-1} \sum_{op} z_{we'op} - e(1 - \sum_{op} (z_{weop} - z_{we-1op})) \leq 0 \ \forall w \in \mathcal{W}, e \in \mathcal{E} \quad (5.37)$$

$$\sum_{e'=e}^{E-1} \sum_{op} z_{we'op} - (E-e)(1 + (\sum_{op} (z_{weop} - z_{we-1op}))) \leq 0 \ \forall w \in \mathcal{W}, e \in \mathcal{E} \quad (5.38)$$

$$t_f - t_e - \sum_{op} DUR_w \Gamma_{pw} (z_{weop} - z_{we-1op} - (z_{wfop} - z_{wf-1op})) \geq - \sum_{e'op} (DUR_w \Gamma_{pw} z_{we'op})$$

$$\forall (f, e) \in \mathcal{E}, f > e, w \in \mathcal{W} \quad (5.39)$$

$$TT - t_e - \sum_{op} DUR_w \Gamma_{pw} (z_{weop} - z_{we-1op}) \geq 0 \ \forall w \in \mathcal{W}, e \in \mathcal{E} \quad (5.40)$$

$$\sum_{o,p} z_{weop} + \sum_{e'=0}^{e} \sum_{op} z_{w'e'op} - (e-1)(1 - \sum_{op} z_{weop}) \leq 1 \forall e \in \mathcal{E}, (w, w') \in \mathcal{W}/PRE_{ww'} = 1 \quad (5.41)$$

$$t_w^i \geq t_e - M(1 + z_{we-1op} - z_{weop}) \quad (5.42)$$

$$t_w^i \leq t_e + M(1 + z_{we-1op} - z_{weop}) \quad (5.43)$$

$$\forall e \in \mathcal{E}, w \in \mathcal{W} / \sum_{w'} (MTL_{w,w'} + MTL_{w',w}) \geq 1$$

$$t_{w'}^i - (ti_w + \sum_{op} \beta_{op} DUR_w \Gamma_{wp}) \leq \Delta \ \forall (w, w') \in \mathcal{W}/MTL_{w,w'} = 1 \quad (5.44)$$

$$\sum_{op} z_{weop} + \sum_{op} z_{w'eop} \leq 1 + M(1 - NONP_{ww'}) \ \forall (w, w') \in \mathcal{W}/NONP_{w,w'} = 1 \quad (5.45)$$

$$\sum_{wp} pz_{weop} \leq num_o^{op} \ \forall o \in \mathcal{O}, e \in \mathcal{E} \quad (5.46)$$

$$\sum_{wop} pz_{weop} AREA_{aw} \leq CAP_a \ \forall a \in \mathcal{A}, e \in \mathcal{E} \quad (5.47)$$

$$z_{ewop} \in \{0,1\} \quad \forall e \in \mathcal{E}, w \in \mathcal{W}, o \in \mathcal{O}, MIN_w^{op} \le p \le MAX_w^{op} \tag{5.48}$$

$$z_{w-1op} = 0 \ \forall w \in \mathcal{W}, o \in \mathcal{O}, MIN_w^{op} \le p \le MAX_w^{op} \tag{5.49}$$

$$\beta_{wop} \in \{0,1\} \ \forall w \in \mathcal{W}, o \in \mathcal{O}, MIN_w^{op} \le p \le MAX_w^{op} \tag{5.50}$$

$$t_e \ge 0 \ \forall e \in \mathcal{E} \tag{5.51}$$

$$t_w^i \ge 0 \ \forall w \tag{5.52}$$

Objective Function The objective function is to minimize the total project cost. The two both possible formulations proposed for the SEE-M are valid for the OOE-M: 5.29 and 5.30.

Order of the events The first event on the project starts at $t = 0$ per constraint 5.31 and at least one task must be active after this event as per, 5.32. In fact, 5.32 is a cut, in order to reduce the solution space. Constraint 5.33 refers to the order of the events, allowing two of them to occur at the same time.

At least one active event per task Each task must be active at least after one of the events, see 5.34, in order to assure the scheduling of all the tasks.

One mode per task We use variables β_{wop}, to select the mode in which each task will be performed. Only one of the variables β_{wop} can be set to 1 per task (5.35) and the tasks can only be performed on the selected mode, as per 5.36.

From now on, we will use *start event* of a task w to refer to the first event e after which w is active. Similarly, the *end event* of task w will be used to refer to the last event after which w is active.

At the same time, most of the following constraints are based on the three values than can take the difference between $z_{weop} - z_{we+1on}$:

- $z_{weop} - z_{we+1op}$= -1. When $e+1$ is the first event after which w is active, so $z_{weop} = 0$ and $z_{we+1op} = 1$.
- $z_{weop} - z_{we+1op}$=1 . When e is the last event after which w is active, so $z_{weop} = 1$ and $z_{we+1op} = 0$.
- $z_{weop} - z_{we+1op}$=0. Otherwise

Continuous processing of each work task Constraint 5.37 means that if task w begins after event e, then it can not be processed before $e-1$. Constraint 5.38 states that if task w ends at event e then w is no longer active $\forall e' \ge e+1$.

Time per task The time difference between w's start event and its end event must be at least the work task's processing time, as per 5.39.

Platform Takt Time The time when a work task ends can be calculated by adding the work tsk processing time to the time at which its start event takes place. Constraint 5.40 means that, in order to cope with the Takt Time restriction, all the work tasks' completion times must be smaller than the Takt Time.

Temporal Constraints If w must precede w', then it must start at an event after which w is no more active as per 5.41.

Maximal time lags are expressed by constraints 5.42 to 5.44. Constraints 5.42 and 5.43 define the start time of a task. This constraints will only be calculated for the tasks involved on maximal time lag constraints. Given 5.42 and 5.43, 5.44 limits the time between the end of a task and its succesor's start time. This time lag is defined as a unique parameter Δ but if needed it can be easily transformed on a parameter dependent on the pair of tasks, $\Delta_{ww'}$.

Non-parallel constraints are 5.45, as two non-parallel tasks cannot be active at the same time.

Ressources Ressource constraints are simpler than for the SEE-M formulation. In this case, only one set of constraints is defined per scarce resource: 5.46 for the quantity of operators per type and 5.47 for the amount of operators per area.

OOE-M Dimensions We will use E' for the number of events used in OOE-M formulation, to distinguish it from the number of events needed in the SEE-M.

This formulation involves:

- $E'W\sum_w((MAX_w^{op} - MIN_w^{op})\sum_o P_{ow})$ event binary variables
- $W\sum_w((MAX_w^{op} - MIN_w^{op})\sum_o P_{ow})$ extra binary variables for the mode selection
- $E' + MTL_w + O$ continuous variables, corresponding to t_e, t_w^i and num_o^{op}

The number of constraints is:

$1 + E' + 2W + WE'(3 + \frac{(E'-1)}{2}\sum_w(MAX_w^{op} - MIN_w^{op})\sum_{wo} P_{ow})+$

$|PRE| + |MTL| + 2MTL_w + |NONP| + OE' + AE'$.

Where: $|\text{PRE}| = \sum_{ww'} PRE_{ww'}$,$|\text{NONP}| = \sum_{w,w'} NONP_{w,w'}$,

$|\text{MTL}|= \sum_{ww'} MTL_{ww'}$ and MTL_w is the number of work tasks for which

$\sum_{w'} MTL_{ww'} + MTL_{w'w} > 0$.

For more detailed information about the number of constraints on OOE-M formulation, see appendix A.

At most $E' = W$. However, we will see that in some cases this number can be reduced. In that sense, it is important to note that the number of constraints is of the order of $(W + WE'^2)$. This means that a reduction in the number of events has greater impact than lowering the number of work tasks.

In terms of W, the number of constraints for this formulation is:

$$W(3+O+A)+W^2[3-\frac{\sum_w (MAX_w^{op}-MIN_w^{op})\sum_{wo} P_{ow}}{2}]+W^3[\frac{\sum_w (MAX_w^{op}-MIN_w^{op})\sum_{wo} P_{ow}}{2}]$$

$$|PRE| + |MTL| + 2MTL_w + |NONP| + 1$$

5.4 Comparison between SEE-M and OOE-M Event Based Multimode Formulations

The OOE-M formulation involves less binary variables than the SEE-M:

$((W+1)(W \sum_w ((MAX_w^{op} - MIN_w^{op}) \sum_o P_{ow}))$ for the event variables and $|NONP|$ for the non parallel constraints.

The number of continuous variables is also fewer on for the OOE-M, with

$(O + A)(W + 1)$ constraints less.

As for the number of constraints, for precedence and maximal time lag constraints, both models are similar. The SEE-M has twice as constraints as the OOE-M for defining the non-parallel constraints. On all, it is not possible to establish a general dominance rule.

Chapter 6
The APSP: Specific Features

On the previous chapter we have presented a new Multimode-Event Based Formulation with generalized temporal constraints. Event Based formulations rely on the fact that, according to Sprecher [Spr95],when a regular measure of performance is concerned, active schedules are the minimal set for optimal solutions. This formulations also take into account that active schedules can be characterized with a limited number of events ($W + 1$ for the SEE formulation and W for the OOE).

However, up to now, Event Based formulations had been applied for the resource constrained scheduling problem (RCSP). We have stated on chapter 4 that the APSP is not a RCSP but a Time Constraint Scheduling Problem.

Also, the previous event based formulations, [KALM11], included only precedence constraints whereas we have included the use of generalized temporal constraints: maximal time lags and non-parallel constraints.

On this chapter we review wether this improvements (the fact that we address a TCSP instead of a RCSP and the introduction of general temporal constraints) have an impact on the two main Event Based Formulation hypothesis: the existance of a regular measure of performance and the minimum number of events needed to guarantee an optimal solution. Section 6.1 reviews the regularity of our objective function. Section 6.2 presents the structure of the active schedules for the APSP, given our objective funtion and finally Section 6.3 justifies the suitability of the chosen number of events.

6.1 Objective Function

In section 3.2 we said that the most classical objective functions are minimising the project Lead Time (LT) or the tardiness ($LT_{min} - LT_{obj}$), together with the project cost ($\sum B_{min} * C_b$). We also presented some other criteria, such as the net present value, resource leveling or resource investment.

From all these, minimising the project Lead Time (LT) or the tardiness ($LT_{min} - LT_{obj}$) are regular measures, whereas objective functions dealing with resource leveling ([NZ98]) and net present value ([Rus70]) are non-regular.

As for TCPS, where the objective function is to minimise the project cost, Möhring, [Moh84] studied it in 1984. He proved that for TCSP the active schedules are also the smallest subset containing optimal solutions. After characterising the objective funtion as a regular measure, he established duality relations with the RCSP.

Later, Guldemond [TG08] refered to a TCSP with regular and overtime time windows, and for that problem he concluded the objective function was not a regular measure.

In order to characterize the APSP objective function, we will refer to the definition proposed by Sprecher [Spr95]: " Given a standard RCSP monomode, for each numerically labelled task w we define a finish time FT_w. A performance measure is then a maping which assigns to each $W-$tuple $(FT_1, ..., FT_w)$ of finish times a performance value $\phi(FT_1, ..., FT_w)$. If ϕ is monotically increasing with respect to componentwise ordering, that is:

$$\begin{aligned} \phi(FT_1, ..., FT_w) > \phi(FT'_1, ..., FT'_w) \text{ implies:} \\ \forall w : FT_w \geq FT'_w \\ \exists j : FT_w > FT'_w \end{aligned}$$

and in addition, minimisation is considered, then we call the performance measure regular."

For the APSP, a solution must assign a set of resources to each work task. Given a feasible solution, for each time instant there will be a number of in progress work tasks, each of them consuming a number of renewable resoures. If we number $1...K$ the available resources from a specific profile, O, each of the active work tasks will use a set of resources $\{k_i..k_j\}$. We will define $R^o_w = 1..K$ as the ordinal of the last resource of type O assigned to task w. Then, our performance measure $\phi = num^{op}_o$ is a regular performance measure as:

$$\begin{aligned} \phi(R^o_1, ...R^o_W) > \phi(R'^o_1, ...R'^o_W) \Rightarrow num^{op}_o > num'^{op}_o \text{ implies:} \\ \forall w : R^o_w \geq R'^o_w \\ \exists w : R^o_w > R'^o_w \end{aligned}$$

Which means that, for reducing the total number of resources needed on the overall project at least one of the tasks has to be done with a lower ranged resource.

In fact, there is a duality relationship between the RCSP and the TCSP. In the RCSP, given a resource availability we wil search the shortest possible Takt Time, whereas in the TCSP we look for the lower resource consumption given a Takt Time. In both cases, the set of optimal solutions are the same. Figure 6.1 represents the minimum number of resources needed for several Takt Times. The first area on the left is the infeasibility area, where no solution is posible for the given takt time. On the middle area, the resource needs decrease as the takt time increases. Finally, on the area on the right, the resource needs are stable no matter how long is the takt time.

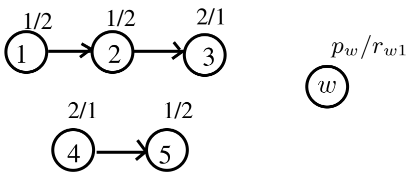

Figure 6.1: Relationship between the Minimum number of resources and the Project Lead Time

6.2 Active Schedules for the APSP

We must remember from section 3.3 that schedules are called semi-active whenever no local left shift is posible for any ot its tasks and active if no local nor global left shift is posible. Moreover, if no local or global shift is posible even considering preemption the schedule is called a non-delay schedule.

Taking as an example the one on Section **??** with five tasks and one resource R_1 (figure 6.2), we will see how the definitions of global / local left shifts, active and semi-active schedules apply to the APSP.

If we consider an instance with a five period platform Takt Time, $TT = 5$, the schedule on figure 6.3 is a feasible schedule. However, *work task* 5 can be done by resources 2 and 3 instead of 3 and 4. Doing this left shift leads to the schedule of figure 6.4. On a next step, *work task* 5 can be further left shifted, and done by ressources 1 and 2. Also, *work task* 4 can have a global left shift, if it is done by resource 1, starting 2 periods later. Those last shifts result on the schedule on figure 6.5, which is an active schedule: work tasks can no longer be left or global shifted in terms of ressource consumption. Note that when a task is left/globally shifted its starting time can be delayed, as far as the takt time restriction is not violated.

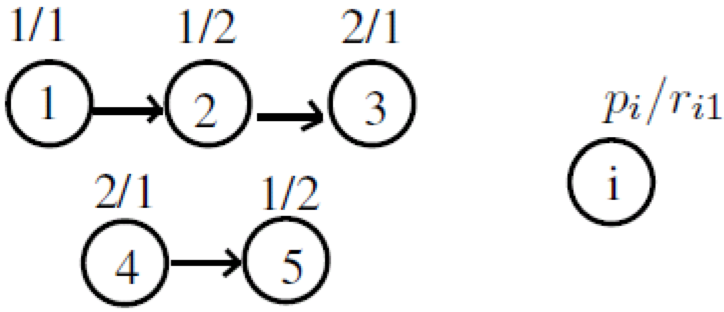

Figure 6.2: RCSP Instance

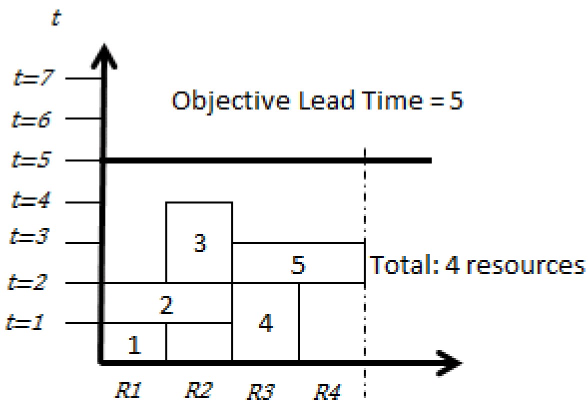

Figure 6.3: Feasible Schedule

Troughout the section, we have refered to a single-mode version of the APSP with only one type of renewable resources for the ease of notation, but the conclusions are equally valable for the multimode and multi-ressource problems. Once each task is assigned to one of its possible modes, the multimode scheedule classification reduces to the single-mode.

6.3 Number of Events

We have already proven that the APSP objective function is a regular measure and therefore, its active schedules are the minimal subset containing the optimal solutions. In this section, we will identify the minimal number of events needed to characterize the active schedules and, as a result, the minimal number of events that needs to be taken into account for solving the APSP.

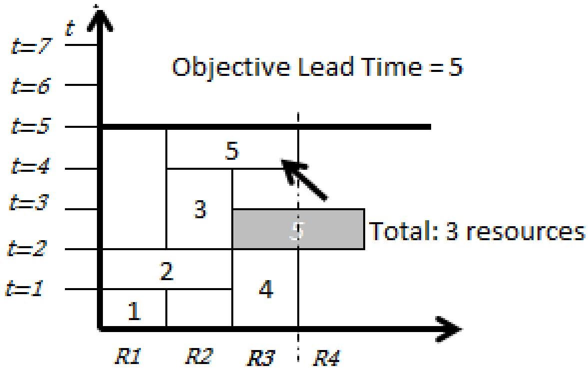

Figure 6.4: Local Left Shift

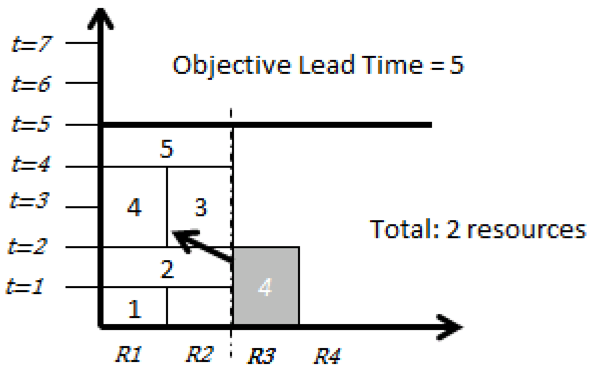

Figure 6.5: Active schedule

We concluded on sections 5.2.2 and 5.3.2 that the number of events has a strong impact on the number of constraints of the model. Therefore, being able to reduce the number of events has a strong impact on improving the performance.

Event based formulations use as an upper bound for the number of events the number of tasks+1 ($E = W + 1$) in the case of SEE and the number of tasks ($E' = W$) in the case of the OOE formulation. In the original RCSP a very limited resource availability or restrictive precedence constraints may lead to a solution where only one task is processed at a time. This would be a poor solution in terms of the project lead time and $E = n + 1$ events will be needed for the SEE formulation, $E = n$ events for the OOE. Solutions with fewer events are possible if two ore more tasks begin at the same event, that is: they can be processed in parallel and begin at the same time with no negative impact on the global project lead time and no precedence / resource consumption violation.

Shall we use as an example the five task scheduling problem from Sections **??** and 6.1 with one resource R_1. From a RCSP approach, depending on the resource availability, several schedules may be feasible (see figure 6.6). Schedule (a) uses as many events as there have been defined, following the $E = W + 1$ / $E' = W$ rule for SEE / OOE formulations. However, we can see that there is another feasible and optimal (in terms of resource consumption) schedule, (b), that uses even fewer events, as tasks 4 and 3 can be done in parallel. Finally, if the resource availability was 3, then there would even be a solution (c) with only 4 events (3 for the OOE formulation).

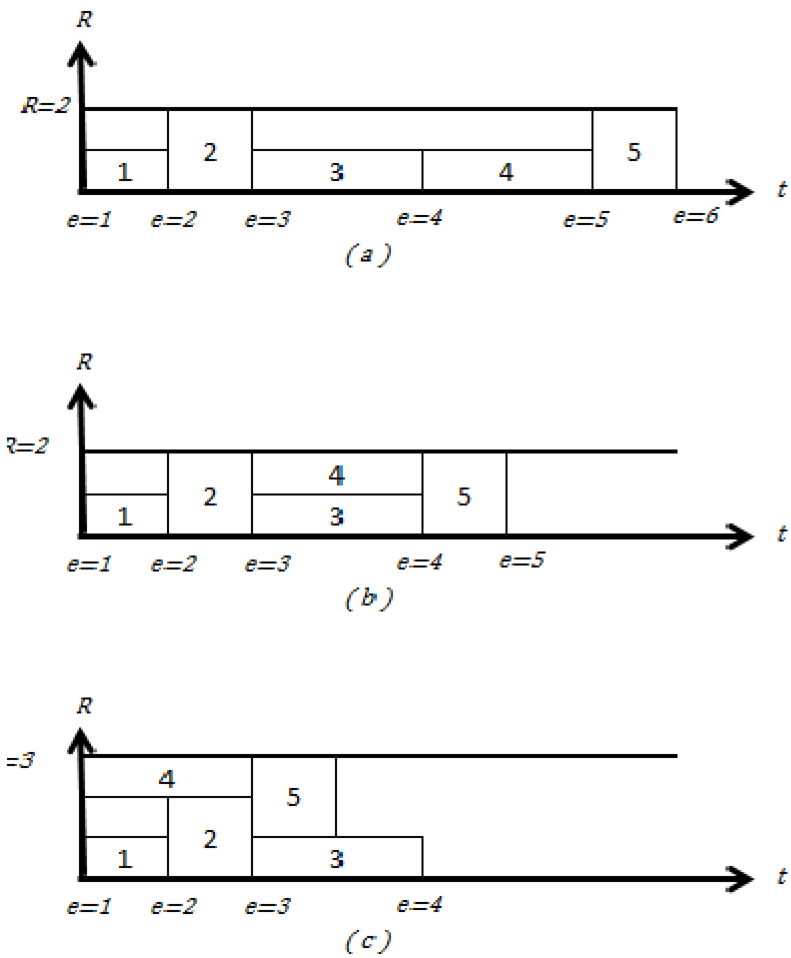

Figure 6.6: Feasible Schedules with Different Number of Events

As we have seen on the previous section and on figure 6.1, given a set of work tasks and precedences, if the optimal solutions given an amount of ressources R will lead to a platform takt time TT then the TCSP with takt time TT will lead to the same optimal solution, with R ressources. In consequence, for a given data set, the number of events to include optimal solutions of the TCSP is the same as the number of events used on the RCSP.

Finally, the original Event Based Model does not deal with non-parallel or maximal/minimal time lag constraints either. Therefore, we will now check the impact on this constraints on the number of events expected for an optimal solution.

Whenever we introduce non-parallel constraints, the minimum number of needed events does not go further than the preliminar $E = W + 1$ ($E' = W$), as the solution with all the tasks in series does not violate any possible non parallel constraints. Anyway, the non-parallel constraints have to be taken into account as a stopper when expecting optimal solutions with less than $E = W + 1(E' = W)$ events. In this case,the number on tasks involved in non-parallel constraints will be a lower bound for the number of events.

As for maximal/minimal time lag constraints, if the time lag is equal to zero, there will be no need of more events. If one task has more than one successors with maximal time lag equal to zero then they will all have the same initial event but their different endings may not lead to a reduction on the number of events needed. In the case of minimal time lags greater than zero, there may be some idle time between tasks, although this does not mean that there must be a greater number of events. On figure 6.7 we can see that because of the minimal time lag between tasks (3) and (4), *event* $e = 4$ is shifted to the left, compared to schedule (a) on figure 6.6 and no more events need to be considered.

It is important to realise that in neither of the models (SEE-M nor OEE-M) the ending time of a task is related to an event. Therefore, we can conclude that for instances with non-parallel and maximal / minimal time lag constraints, the number of events defined by the original model (SEE: $E = W + 1$, OEE $E = W$) is enough for an optimal solution.

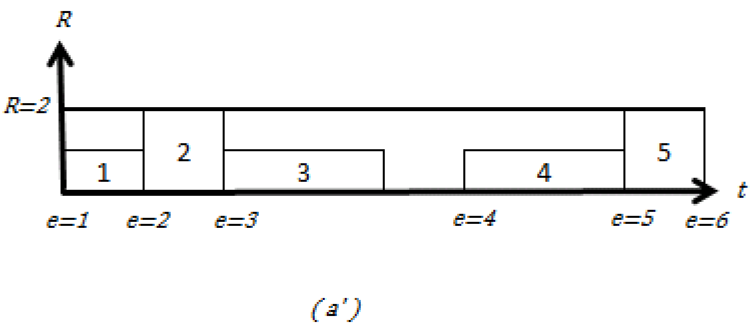

Figure 6.7: Impact of MTL Constraints

Chapter 7
The APSP: Experimentation

7.1 Implementation

The computational results were obtained using CPLEX12.4 solver. The tests were carried out on an Intel-Core i7-2630QM processor with 2GHz and 4 GB RAM, running Windows 7.

The AIMMS modeling environment was used for the implementation of the MILP formulation. As well as this, an interface for input data via Excell was developped. Although not estrictly necessary for the initial experimentation, it is an enabler for future real-world instance solution.

As well as this, AIMMS was used for a graphical result output, in the shape of a Gantt diagram (figure **??**)

7.2 Instances

As the standard PSPLIB instances are not valid for the structure of the problem, four new sets of 8 task instances were used. Table 7.1 summaries the main instance characteristics. Sets 3 and 4 were extended in order to create instances of up to 11 tasks. Their characteristics are listed on table 7.2.

On all, 75 different combinations of data sets, number of tasks, Takt Time and

Table 7.1: Instance Characteristics

Set	Nº Tasks	Precedences	$\sum MTL$	$\sum NONPL$	Op. Profiles	Areas	Nº Modes
Set1-8	8	6	1	1	2	2	12
Set2-8	8	8	1	1	2	2	16
Set3-8	8	7	1	1	2	2	17
Set4-8	8	7	1	1	2	2	16

Table 7.2: Extended Instance Characteristics

Set	Nº Tasks	Precedences	$\sum MTL$	$\sum NONPL$	Op. Profiles	Areas	Nº Modes
Set3-9	9	8	1	1	2	2	18
Set3-10	10	9	1	1	2	2	18
Set3-11	11	10	1	1	2	2	18
Set4-9	9	8	1	1	2	2	16
Set4-10	10	9	1	1	2	2	16
Set4-11	11	10	1	1	2	2	16

number of events were tested for each formulation. On this chapter we will comment on the main conclusions related to this results. The detailed instances are also availablle on [ASP] and the whole computational results are on appendix B.

7.3 Computational Results

7.3.1 SEE-M Results

All instances were solved up to optimality, taking times from seconds to fifteen minutes. For each set and Takt Time, different number of events where tested. Starting from the theoretical minimum number of events, they were reduced until solutions where no longer optimal. As it was expected, following the conclusions on sections 5.2.2 and 5.3.2, the solution time decreased exponentially with the number of events, even when solving the same set of instances. The first Set (Set1-8), when solved for a $TT = 11,5$ days took from 2 to 281 seconds, depending on the number of events. The harder instance to solve was Set3-11, that took up to 4133 seconds for a $TT = 17$ days and 11 events.

Figure 7.1 shows the evolution of the SEE-M solution times for a same instance and input Takt Time whenever the number of events was changed. On this figure the series data include information on the data set, the number of tasks and the input Takt Time: Set1-8-11.5 stands for the solution of data set 1, with 8 tasks and a TT=11.5 days.

The event reduction not only lowers the number of variables but also reduces the integrality gap, as it defines a tighter polyhedra. In some cases, the integrality gap is reduced by improving the First Lower Bound (FLB) but also the First Integer Solutions (FIS). However, the improvement on the First Integer Solution does not always happen. For example, Set2-8 with $TT = 33$, if solved with 7 events has a First Lower Bound $FLB = 1,84$ and First Integer Solutions $FIS = 7$. If for the same set we use 10 events, the first solutions are $FLB = 0,29$ and $FIS = 10$. In fact, with 10 events, the number of explored nodes is 136 times bigger than with 7

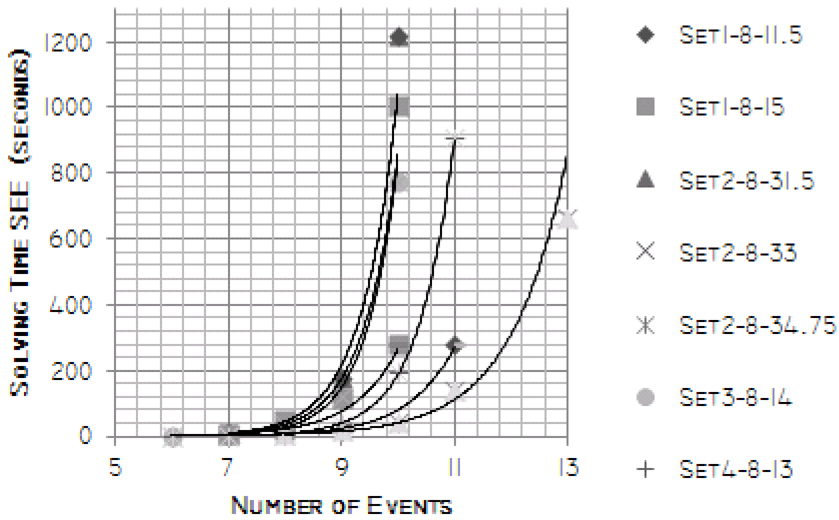

Figure 7.1: SEE-M Solution Time Vs number of Events

events (405935 vs 2981). The relationship between the number of events and the FLB is plotted on figure 7.2, using a the same set nomenclature as figure 7.1.

Table 7.3 includes the main information to measure the performance during the scheduling of the 8 tasks sets. Columns 1 to 3 are related with the input data: the first column is used to identify the set of instances, where $Set1-8$ stands for the first data set with 8 tasks. The second column shows the input Takt Time. Finally, the third column is the number of events defined for the formulation. Columns 4 to 8 refer to the results. The "Nodes" are the number of nodes explored on the Branch & Bound until the optimal solution. The first lower bound (FLB) and first integer solution (FIS) include the value of this first solutions. As for the Solution t, it is the time to provide the optimal solution. The last column is the optimal number of resources.

As for the evolution of the solution time throughout different Takt Times, on average the solution time also grew as the objective Takt Time got closer to the Critical Path Lead Time. On table 7.4, we can see the solution time for Set2-8 instance and different Takt Times.

Finally, focusing on the influence of the number of tasks, most of the instances required more solution time with the same number of events when new tasks are

Table 7.3: SEE-M Impact Number of Events

Instance	TT	Events	Nodes	FLB	FIS	Solution t (sec)	Optimal Sol
Set1-8	11,5	7	1546	2,3	10	1,97	8
Set1-8	11,5	9	11203	1,92	9	21,09	8
Set1-8	11,5	11	105195	1,63	11	281	8
Set1-8	15,5	7	3777	2,3	10	7	6
Set1-8	15,5	8	13203	1,75	9	17,32	6
Set1-8	15,5	10	219403	1,12	9	282,97	6
Set2-8	31,5	7	7069	1,85	9	10,2	7
Set2-8	31,5	8	31116	1,194	9	39,76	7
Set2-8	31,5	9	119162	0,64	8	176,98	7
Set2-8	31,5	10	511955	0,29	10	1215,95	7
Set2-8	33	7	2981	1,84	7	6,65	6
Set2-8	33	8	25778	1,194	8	46,49	6
Set2-8	33	9	45205	0,64	9	116,81	6
Set2-8	33	10	405935	0,29	10	1007,87	6
Set2-8	34,75	8	1912	1,19	10	3,39	5
Set2-8	34,75	9	6926	0,64	11	19,66	5
Set2-8	34,75	10	10619	0,296	9	44,69	5
Set2-8	34,75	11	28112	0,07	8	142,52	5
Set2-8	34,75	13	75825	0,01	10	663,27	5
Set3-8	14	6	472	2,8	8	0,61	6
Set3-8	14	7	4631	1,7	8	6,35	6
Set3-8	14	8	23407	1,13	7	34,4	6
Set3-8	14	9	94129	0,69	7	121,98	6
Set3-8	14	10	506132	0,36	8	776,76	6
Set4-8	13	6	112	3,05	7	0,3	7
Set4-8	13	7	1957	2,14	8	2,2	7
Set4-8	13	8	3341	1,43	8	6,85	7
Set4-8	13	9	29076	0,92	9	36,58	7
Set4-8	13	10	127109	0,55	9	193,25	7
Set4-8	13	11	422579	0,26	9	905,13	7

Table 7.4: Sample SEE-M solving time for different Takt Times

Instance	$LT = 31.5$	$LT = 33$	$LT = 34.75$	$LT = 41$
Set2-8	10.2s	6.65s	1.79s	0.83s

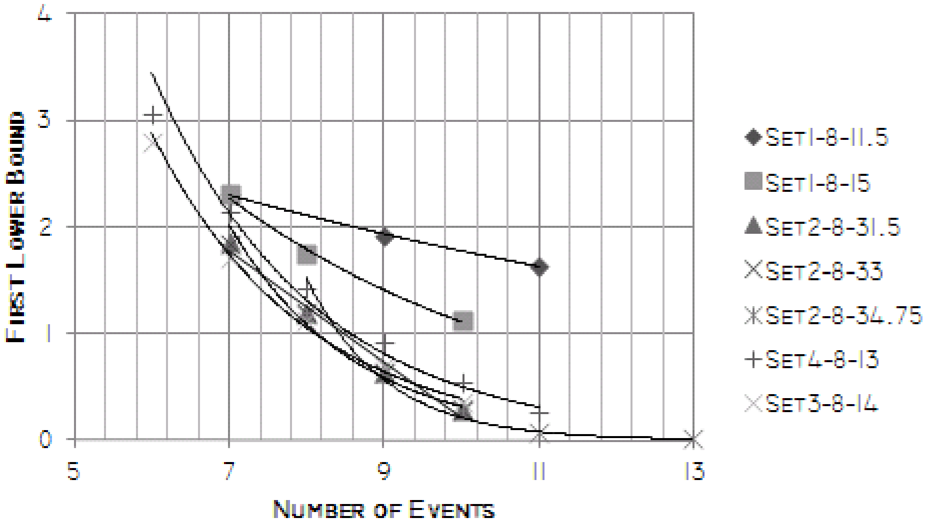

Figure 7.2: SEE-M First Lower Bound Vs Number of Events

added. Withal, some were solved faster with more tasks. This shows that in some cases the structure of the problem is more important than the number of tasks itself. We also stated that the hardening of the instances as we add new tasks is wider on the cases were we are using more events.The relationship between the solution time and the number of tasks has been plotted on figure 7.3. Here, each series of data represents the solution of a same set with a fixed number of events adding different number of tasks: Set3-8E states for the third Set solved with 8 Events and 8 to 11 tasks.

7.3.2 OOE-M Results

The same instances were used for the computational study of OOE-M formulation. All instances were solved up to optimality. In this case, they took times from 0,1 seconds to eight minutes. As happened with the SEE-M formulation, the solution time decreased exponentially with the number of events, even when solving the same set of instances. For example, Set3 with 8 tasks (Set3-8) and $LT = 14$ days, was solved in times from 0.2 to 181 seconds. The harder instance to solve was the eleven tasks Set3 (Set3-11), which tool 452 seconds for $TT = 17$ and 11 events. This was also the harder instance for the SEE-M formulation.

Figure 7.4 shows the evolution of the OOE-M solution times for a same instance

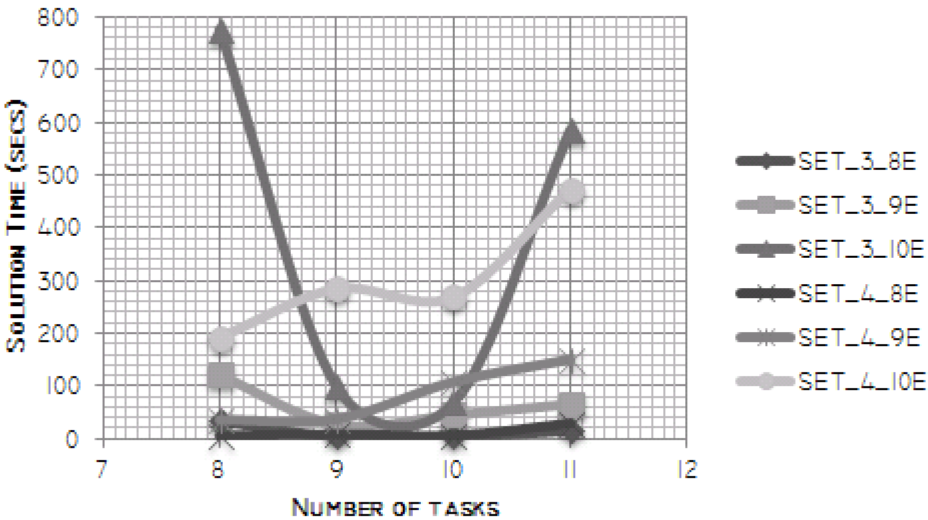

Figure 7.3: SEE-M Solution Time vs Number of Tasks

and input Takt Time whenever the number of events was changed. On this figure, the series data include information on the data set, the number of tasks and the input Takt Time: Set1-8-11.5 stands for the solution of data set 1, with 8 tasks and a TT=11.5 days.

As in the SEE-M formulation, the First Lower Bound increased when lowering the number of events at the same time that the number of nodes to be visited before the optimal solution became lower. We will take as an example Set3 when tested with $TT = 14$ and 5 to 10 events. The FLB varied from 3,8 to 1,47 and the number of visited nodes grew from 72 to 261475. The relationship between the number of events and the first LB is plotted on figure 7.5, where the series of data are numbered as in figure 7.4

Table 7.5 includes all the data about nodes, First Lower Bound (FLB), First Integer Solution (FIS), solution time and optimal solution for the four sets with 8 tasks when solved for different Takt Times and number of events.

As for the evolution of the solution time throughout different Takt Times, on average the solution time also grew as the objective Takt Time approached to the Critical Path Lead Time. On table 7.6, we can see the solution time for Set2-8 instance for different Takt Times:

Table 7.5: OOE-M Impact Number of Events

Instance	TT	Events	Nodes	FLB	FIS	Solution t (sec)	Optimal Sol
Set1-8	11,5	7	478	2,4	10	0,67	8
Set1-8	11,5	9	1918	2,29	10	1,93	8
Set1-8	11,5	11	10688	2,22	10	12,68	8
Set1-8	15,5	7	857	2,2	10	0,8	6
Set1-8	15,5	8	1385	1,92	6	1,33	6
Set1-8	15,5	10	11416	1,525	10	10,72	6
Set2-8	31,5	7	3561	3,6	8	4,66	7
Set2-8	31,5	8	10998	3,5	7	10,58	7
Set2-8	31,5	9	18506	3,43	7	20,05	7
Set2-8	31,5	10	48674	3,38	7	61,11	7
Set2-8	33	7	678	3,6	6	0,83	6
Set2-8	33	8	2852	3,5	7	2,65	6
Set2-8	33	9	10683	3,43	7	10,3	6
Set2-8	33	10	7793	3,34	6	9,63	6
Set2-8	34,75	7	141	3,6	7	0,47	5
Set2-8	34,75	8	93	3,5	7	0,58	5
Set2-8	34,75	9	630	3,43	6	1,45	5
Set2-8	34,75	10	276	3,34	6	1,11	5
Set2-8	34,75	11	177	3,33	7	1,14	5
Set2-8	34,75	13	150	3,3	5	1,45	5
Set3-8	14	5	72	3,8	7	0,22	6
Set3-8	14	6	837	2,58	7	0,87	6
Set3-8	14	7	2207	2,14	7	2,14	6
Set3-8	14	8	8593	1,85	7	8,24	6
Set3-8	14	9	21865	1,64	7	19,31	6
Set3-8	14	10	261475	1,47	8	181,07	6
Set4-8	13	5	0	4,74	7	0,11	7
Set4-8	13	6	408	2,5	9	0,65	7
Set4-8	13	7	1819	2,14	8	2,06	7
Set4-8	13	8	3460	1,88	7	3,82	7
Set4-8	13	9	16935	1,67	9	17,18	7
Set4-8	13	10	53020	1,5	8	45,66	7
Set4-8	13	11	25773	1,36	8	29,28	7

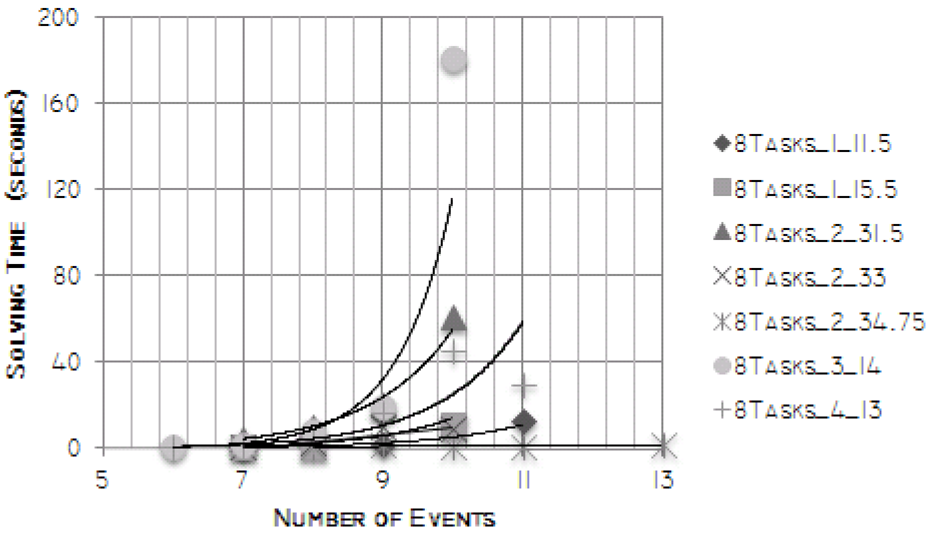

Figure 7.4: OOE-M Solution Time Vs number of Events

Table 7.6: Sample OOE-M solving time for different Takt Times

Instance	$LT = 31.5$	$LT = 33$	$LT = 34.75$	$LT = 41$
Set2-8	4.66s	0.83s	0.47s	0.2s

To end with, most of the instances required more solution time with the same number of events when new tasks are added. Withal, some were also solved faster with more tasks. As well as this, the hardening of the instances as we add new tasks is wider on the cases were we are using more events.The relationship between the solution time and the number of tasks has been plotted on figure 7.6. As on figure 7.3,each series of data represents the solution of a same set with a fixed number of events adding different number of tasks: Set3-8E states for the third Set solved with 8 Events and 8 to 11 tasks.

7.3.3 SEE-M and OOE-M Comparative Study

By comparing sections 7.3.1 and 7.3.2, we can see that both models have similar behaviour as far as the impact of variations on the number of events, number of tasks and Takt Time on the solution time.

The results in terms of solution time, number of nodes and first lower bound have

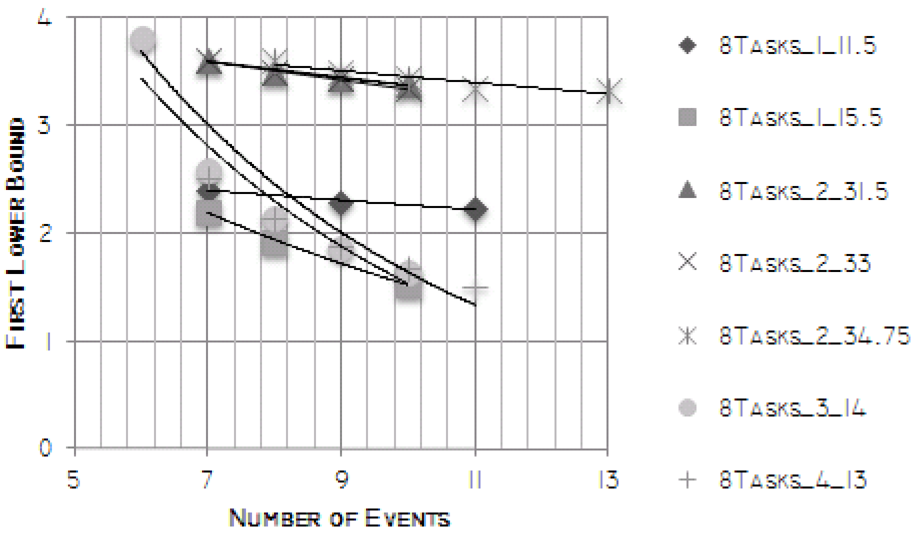

Figure 7.5: OOE-M First Lower Bound Vs number of Events

been better for all the instances with the OOE-M formulation than with the SEE-M formulation. Figure 7.7 shows the histogram for the division of the time spended for a solution with the OOE-M formulation between the time spended by the SEE-M formulation.The only two cases where the solution time is longer for the OOE-M formulation are instances with solution times within the range of 0,5 seconds, where the absolute difference is not significant.

Table 7.7 compares all the performance measures for several cases on the SEE-M and OOE-M formulations. There, it is easy to see that the difference between both formulations grows with the number of events. Therefore, as the complexity of the instances grows the use of the OOE-M formulations becomes more and more suitable. It is also important to rememember that, although the comparison has been made between instances with the same number of events, the OOE-M formulation is always capable of calculating a solution using one event less than the SEE-M formulation. As a result, the performance difference between both formulations is even wider.

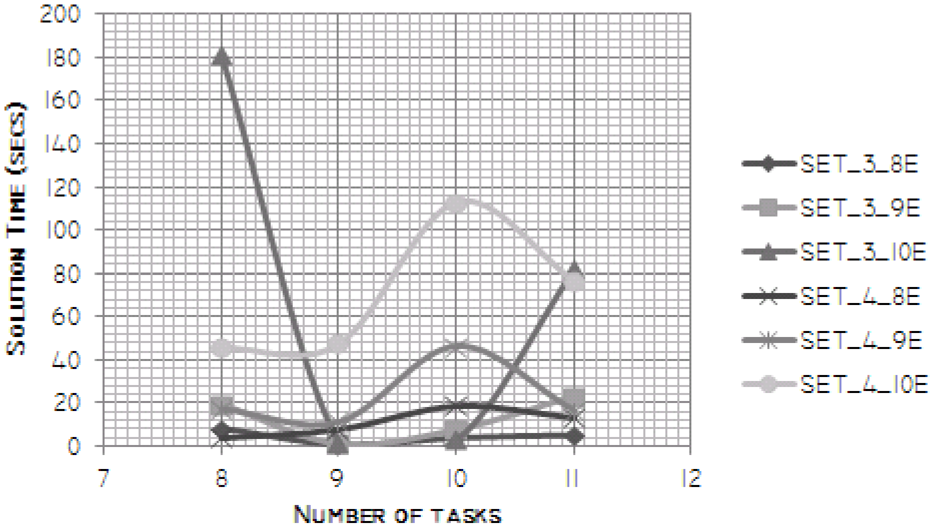

Figure 7.6: OOE-M Solution Time vs Number of Tasks

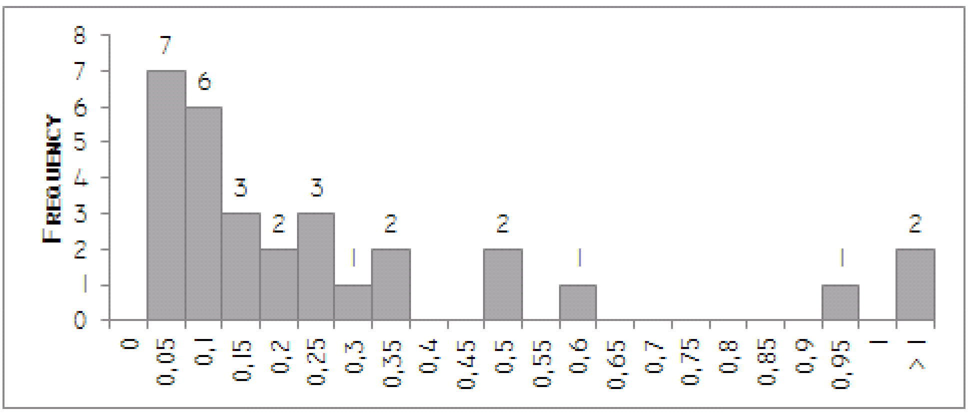

Figure 7.7: OOE-M Solution Time / SEE-M Solution Time

Table 7.7: SEE-M and OOE-M performance comparison

Set-Tasks-LT-Ev	FLB SEE-M	FLB OOE-M	FIS SEE-M	FIS OOE-M	Nodes SEE-M	Nodes OOE-M	Sol. t SEE-M	Sol. t OOE-M
Set1-8-11,5-7	2,3	2,4	10	10	1546	478	1,97	0,67
Set1-8-11,5-9	1,92	2,29	9	10	11203	1918	21,09	1,93
Set1-8-11,5-11	1,63	2,22	11	10	105195	10688	281	12,68
Set1-8-15,5-7	2,3	2,2	10	10	3777	857	7	0,8
Set1-8-15,5-8	1,75	1,92	9	6	13203	1385	17,32	1,33
Set1-8-15,5-10	1,12	1,525	9	10	219403	11416	282,97	10,72
Set2-8-31,5-7	1,85	3,6	9	8	7069	3561	10,2	4,66
Set2-8-31,5-8	1,194	3,5	9	7	31116	10998	39,76	10,58
Set2-8-31,5-9	0,64	3,43	8	7	119162	18506	176,98	20,05
Set2-8-31,5-10	0,29	3,38	10	7	511955	48674	1215,95	61,11
Set2-8-33-7	1,84	3,6	7	6	2981	678	6,65	0,83
Set2-8-33-8	1,194	3,5	8	7	25778	2852	46,49	2,65
Set2-8-33-9	0,64	3,43	9	7	45205	10683	116,81	10,3
Set2-8-33-10	0,29	3,34	10	6	405935	7793	1007,87	9,63
Set2-8-34,75-8	1,19	3,5	10	7	1912	93	3,39	0,58
Set2-8-34,75-9	0,64	3,43	11	6	6926	630	19,66	1,45
Set2-8-34,75-11	0,07	3,33	8	7	28112	177	142,52	1,14
Set2-8-34,75-13	0,01	3,3	10	5	75825	150	663,27	1,45
Set3-8-14-6	2,8	2,58	8	7	472	837	0,61	0,87
Set3-8-14-7	1,7	2,14	8	7	4631	2207	6,35	2,14
Set3-8-14-9	0,69	1,64	7	7	94129	21865	121,98	19,31
Set3-8-14-10	0,36	1,47	8	8	506132	261475	776,76	181,07
Set4-8-13-6	3,05	2,5	7	9	112	408	0,3	0,65
Set4-8-13-8	1,43	1,88	8	7	3341	3460	6,85	3,82
Set4-8-13-10	0,55	1,5	9	8	127109	53020	193,25	45,66
Set4-8-13-11	0,26	1,36	9	8	422579	25773	905,13	29,28

This comparative results are coherent with the results of Koné [KALM11] for the mono-mode case with only precedence constraints, who concluded that the OOE outperformed the SEE formulation for all the instance sets. As stated by [Tsa06], a good formulation must have two characteristics: its tightness and its compactness. The tightness refers to the quality of its approximation to the convex hull of integer feasible solutions and the compactness is a measure of the number of variables and constraints that are used to formulate a given problem. The OOE-M is better than the SEE-M formulation in both aspects.

7.4 APSP Instance classification

Throughout all section 6.3 we have outlined some of the problem characteristics that can have a positve impact on reducing the number of needeed events (and therefore the calculation time) for achieving an active schedule and an optimal solution. We have identified that the posibility of parallel procesing is important, together with the possibility of the syncronisation of the beginning of work tasks.Also, we have commented on the impact of common maximal/minimal time lag constraints.

Afterwards, from the computational results we have identified that different sets of instances with apparently the same complexity may require very different solution times. In this section, we will go deeper into this characteristics and identify some indicators in order to classify the instances with respect to the number of events needed and the hardness of their solution.

On the one hand, one of the enabling characteristics for reducing the event number is the capability of processing tasks in parallel, which is linked to the precedence-oriented instance indicators. On the other hand, the hybrid indicators have the desirable feature of providing a more global characterization of the instance hardness.

Therefore, we will review the precedence oriented indicators and their impact and then we will focus on only one time featured indicator and several hybrid (time and resource) featured ones.

We will go back to the common RCSP intances indicators explained on section 3.7 . Those indicators have been constructed for the RCSP. However, as the solution space we are looking at is the same, we can use them as a starting point for defining more suitable ones for the APSP.

7.4.1 Precedence-Oriented indicators

The Precedence Graph structure is directly related to the capability of processing tasks in parallel. The characteristics related to the precedence graph have been until now studied by the precedence-oriented instance indicators from section 3.7.1.

The *serial or parallel indicator* is useful for knowing the possible existing parallel paths. This information, can be completed by the *activity distribution indicator* , in order to have a hint on the interaction between the paths.

Moreocer, the *long path* , *short path* and *topological float* will give information on parallel processing: a long path with a high number of short paths will have more opportunities for parallel processing once scheduled the dominant long path. As well as this, the*topological float* provides us with information about the flexibility on each activity. However, these three last indicators include accurate information only if the activity length is homogeneus, as on the other case, they will be seriously biased.

All the previous indicators can be directly used for APSP instances.

7.4.2 Time Featured indicator

Takt Time Free Float Ratio LT-FFR

On section 3.7.3 we defined the Free Float Ratio of a task using the difference between the maximum and minimum task start times that enabled the project to be completed on its earliest precedence-feasible lead time.

In our case, we will calculate the Free Float Ratio with respect to the platform Takt Time. For activities with multiple modes involving different durations, we will choose the maximum resource consumption per activity. We will name this indicator the Takt Time Free Float Ratio in order to distinguish it from the original one:

$$TT - FFR = \frac{\sum_w D_w^{min}}{\sum_w D_w^{min} + CFF_w)}$$

Where $D_w^{min} = Gamma_{MAX_w^{op} w} D_w$ is the minimal duration of work task w, the one with most resource consumption. CFF_w is the Constrained Free Float, or the difference between its latest and earliest start time for a total platform duration equal to the given Takt Time.

This indicator is useful to know the improvement capability once a feasible schedule is identified. A TT-FFR equal to 1 means that no task has any buffer and must be scheduled at its earliest posible date and with its maximal resource consumption in order to comply with the Takt Time. The lowest the TT-FFR, the most flexibility of the project. We would expect an impact on the solution time following a easy-hard-easy bell shape, as for instances with low flexibility there are few solutions to study and for those with high flexibility it is easier to find one of the optimal solutions.

7.4.3 Hybrid Indicators

Takt Time Strength (TTS)

This indicator is equivalent to the Resource Strength (RS). As explained on section 3.7.4, this indicator acts as a scale between the minimum resource demand and the maximum peak demand for the earliest feasible schedule:

$$B_k = B_k^{min} + RS(B_k^{max} - B_k^{min})$$

The equivalent for the APSP will be:

- B_k is the available resource quantity of the project. Instead of it, we will use the platform Takt Time, TT
- B_k^{min} is the minimum resource quantity required for a feasible schedule, no matter the Lead Time. We will use for this the minimum platform Takt Time, using the minimal duration (maximal resource consumption) per task: TT^{min}
- B_k^{max} is the maximal peak demand in the case of the earliest precedence-schedule, that is, a schedule that takes into account only the precedence constraints in order to minimize the objective function. In our case, we will calculate the maximal Takt Time, for the case where the resource consumption is the minimal possible for each type, this is, the takt time for a schedule where $B_k = B_k^{min}$. We will name it TT^{max}

$$TT = TT^{min} + LTS(TT^{max} - TT^{min})$$
$$TTS = \frac{TT - TT^{min}}{(TT^{max} - TT^{min})}$$

For the cases where $LTS > 1$ the instance is easy to solve, as the available Takt Time is more that the one achieved with the least posible resources.

One of the drawbacks of the RS is that once a task demands the whole resource availability ($B_k = B_k^{min}$), $RS = 0$ independently from the rest of the demands. In our case, if one of the paths demands the whole Takt Time available then the LTS will be also equal to zero. Therefore, it must be suitable to combine this indicator with the I_2 (Serial or Parallel Indicator).

Takt Time Disjunction Ratio (TTDR)

The Disjunction Ratio (section 3.7.4) provided information on the pairs of activities that could not be performed in parallel due to resource constraints. This is equivalent to measuring the number of minimal forbidden sets or, even better, the number of unsolved minimal forbidden sets.

In our case, we can define the minimal forbidden sets as the paths where at least one of the task needs to be assigned more than the minimal resources in order to comply with the Takt Time. For each of the paths, we will have to choose between all the path tasks to decide which task to shorten, as on the resource minimal forbidden sets there was a decision to be made on the precedence constraint to add.

To count the number of needed decisions we will go back to section 3.7.1. There we defined the progressive level and regressive level per task which measure, in terms of number of tasks, the distance from a task to the start node (progressive level) or end node (regressive level). Each path ends on a task whose regressive level is $RL = m$.The number of possible tasks to be re-assigned equal to the progressive level of that last task.

Also, we will use the same maximum time schedule as for the calculation of the Takt Time Strength (TT^{max}). The end time of a task w on that schedule will be $LS_w^{TT^{max}}$. We will define the set $LATE_w$ for those tasks for which $RL_w = m$ and $LS_w^{TT^{max}} > TT$

$$LTDR = \frac{\sum_{w \in LATE_w} PL_w}{n}$$

This indicator measures the number of tasks involved in paths that must be shortened divided by the total number of work tasks. It can be bigger than one, as a work task can be involved in more than one 'long' path. The higher the indicator, the more work tasks that must be shortened and therefore the hardest the instance will be to solve.

Chapter 8
Conclusions & Further Research

Prior to this work, we had identified a knowledge gap related to the RCSP applicable for the scheduling of aeronautical assembly plaforms (or, the Aeronautical Platform Scheduling Problem, APSP). After a deep litterature review, where we have gone through the main existing models for the more general RCSP, we have chosen to develop an Event Based Formulation for our problem. To do so, we have dealt with general temporal constraints, maximal and minimal time lags and multimode scheduling. This problems had been rarely addressed on their own in the existing litterature and no references have been found dealing with them together. Due to this features, the two new formulations we have developped are a contribution not only for the aeronautical industry but also for general scheduling.

The computational results from chapter 7 have proven that the model is able to solve up to optimality small instances. As well as this, they have enabled as to make a comparative study between the two event based formulations: SEE-M and OOE-M formulations. The results of this comparations are coherent with the ones reported by Koné for the single mode with only precedence constraints model, [KALM11].

However, in order to extend it to bigger instances, it is necessary to improve the solver performances. Both SEE-M and OOE-M formulations have a number of constraints that grows as $(\mathcal{O}(E^2W)$ where E is the number of events and W the number of work tasks to be scheduled. From the computational results we know that most of the instances can be solved with less events than the established upper bound of $W+1$ for SEE-M and W for OOE-M. Therefore, the use of pre-processing to calculate the real needed number of events will lead to major performance improvements. In that sense, we have already defined a set of indicators that will help us for the characterisation of APSP instances and as a result, to the developpment of more suitable and efficient pre-processing techniques.

On a next step, we will use those indicators for a more accurate definition of the number of events. At the same time, we will also improve the solution procedures by the introduction of decomposition or column generation techniques, as well as hybrid procedures including the most common scheduling heuristics and metaheuristics.

Nevertheless, the improvement of our model's performance is not the only research direction we have identified. Another series of research questions have arisen. Although they come from in the context of the aeronautical industry, they can be also extended to other manufacturing scheduling problems:

Can all the Aeronautical Plaform Scheduling Problems be solved with the same techniques?

As standarization is more and more frequent on all the activities of a company, scheduling methods and tools tend also to be common to all its sites and even divisions. In that sense, the assembly lines corresponding to different products (Final Assembly Lines, major component and small parts manufacturing) would normally share the same scheduling method.

Although this has advantages for knowledge sharing, there can be a risk of treating alike problems of different nature. It has been proved, for example [KALM11], that different kind of scheduling instances may lead to different lower bound quality and exact solution times with a same solution method. One of the objectives of the research will be to identify the instances Key Characteristics that can lead to the use of different solution approaches.

Can preliminary design be focused on scheduling?

Since the 1980s, concurrent engineering has been well studied and widely implemented in industry to compress time to market and cost. The basic concept is to reduce design-loops and include all life cycle requirements within a first product development, using parallel engineering. As a result, manufacturing requirements are taken into account form the preliminary design stages. Ma [MCT08] identified and analyzed some research issues with respect to information integration and sharing for future concurrent and collaborative engineering. However, on the assembly phase they focus on the technical feasibility of the assembly while the scheduling of the assembly is also impacted by the product design and taking it into account in a early stage can enable the profitable industralization of a project.

For scheduling not to be left apart, there is a need of clear rules to anticipate the impact on scheduling of product development decissions. This can only be achieved if there are some rules which can help the concurrent engineers to predict how different designs may behave as far as scheduling is concerned. In addition, it has to be taken into account that those conclusions must be stated in a way that non-scheduling experts can take advantage of them.

Is continuous time the best approach?

As far as time modeling is concerned, we have chosen a continuous time formulation because of its better performance with our large time horizons.

However, on the industry time is not always continuous. Works to be begun just before the end of a working period are usually delayed until the beginning of the next journey, just because of human factors or because of non-work preemption. A continuous time model implicitly assumes that the work can be re-started on the next journey as if it had never been interrumpted.

A common approach to this problem is to obtain a valid continuos time schedule and to deal with the real working period restrictions afterwards.

One question to be answered is how much better could be the real solution to be implemented if it had from the beginning a real time approach. This is, an approach that takes into account under which time restrictions a new task can be assigned to a existing free resource. The alternative solution may be similar to the continuous time one or significantly different. The cost of this decision seems not to have been addressed yet in the litterature.

Once deviations occur, which is the best re-scheduling method?

Although the aim of aeronautical industry is to have standart schedules with zero variation between products, deviations occur. Once the Platform tasks have been scheduled, the solution may be subject to modifications during the real production phase. At this moment, the inefficiencies or non-conformities (both of the process and its supply chain), may make it impossible to comply with the ideal planning, that needs to be adapted to the new specific scenario.

Due to point of use deliveries and production planning, a valid re-scheduling must be as alike as possible to the standard one. In that sense, re-scheduling is also a scheduling problem and may need of as complex methods and tools as the baseline (ideal) scheduling. It has to deal with all the restrictions of the previous model, but it also has to deal with additional ones coming from the real state of the line, different resource availability and different objective function: getting back to the standart path may be more important than labor costs, for example.

Another specific characteristic of the re-scheduling approach is the performance needs: An original solution may be prepared on a monthly / year basis, being more important the optimality of the solution than the calculating time. Re-scheduling will require a more agile answer, as production needs not be stopped.

If scheduling is normally addressed with heuristics in the industry, re-scheduling is normally done relying on human knowledge and it has not been dealt with in the litterature.

To sum up, four additional research questions have been proposed. The two first ones *instance key characteristics* and *design focused on scheduling* have to do with the study of the characteristics of the APSP. By classifying them, good design practices and suitable solution methods will be identified. To do this, the contribution of this first work has been to provide a new set of indicators valid for the characterization of the APSP.

The two MILP models we have formulated and solved are a starting point for the other two research questions. The *time modeling* issue can be addressed by defining on the existing model different time windows for each shift and penalizing the use of differente time windows for a same work task. For the *re-scheduling*, the event based formulations must be extended but can be used as a basis. Also, the current scheduling model will be useful to evaluate the difference between re-scheduling from zero (not taking into account the first solution) or looking forward to minimising the changes.

References

[aim] Aimms. http:
www.aimms.com.

[AKL+10] Christian Artigues, Oumar Koné, Pierre Lopez, Marcel Mongeau, Emmanuel Néron, and David Rivreau. *Benchmark Instance Indicators and Computational Comparison of Methods*, chapter 7, pages 107–135. ISTE, 2010.

[Alt10] Hans-Henrich Altfeld. Lean manufacturing in the aircraft industry. Royal Aeronautical Society, 2010.

[AMR03] Christian Artigues, Philippe Michelon, and Stéphane Reusser. Insertion techniques for static and dynamic resource-constrained project scheduling. *European Journal of Operational Research*, 149(2):249 – 267, 2003. <ce:title>Sequencing and Scheduling</ce:title>.

[aru] Arum european proyect. http://www.infodefensa.com/?noticia=eads-se-une-al-proyecto-europeo-arum-que-facilitara-la-produccion-aeroespacial. Accessed: 2012-06-02.

[ASP] Asps. https://www.dropbox.com/sh/9070da1dz6typ2o/V0jCAtR_8y.

[AV93] Tamarit J.M. Alvarez-Valdes, R. The project scheduling polyhedron: Dimension, facets and lifting theorems. *European Journal of Operational Research*, 67:204–220, 1993.

[BDM+99] Peter Brucker, Andreas Drexl, Rolf Möhring, Klaus Neumann, and Erwin Pesch. Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112(1):3 – 41, 1999.

[BK11] P. Brucker and S. Knust. *Complex Scheduling*. GOR-Publications. Springer, 2011.

[BKFS92] Wolfgang W. Bein, Jerzy Kamburowski, Matthias F, and M. Stallmann. Optimal reduction of two-terminal directed acyclic graphs. *SIAM Journal on Computing*, pages 1112–1129, 1992.

[BP00] Philippe Baptiste and ClaudeLe Pape. Constraint propagation and decomposition techniques for highly disjunctive and highly cumulative project scheduling problems. *Constraints*, 5:119–139, 2000.

[CA57] Ackoff R. L. Churchman, C. W. and E. L. Arnoff. *Introduction to Operational Research.* John Wiley and Sons, New York, 1957.

[CA10] Emmanuel Neron Christian Artigues, Sophie Demassey. *Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications.* Wiley Library, 2010.

[Chr87] Alvarez-Valdes R. Tamarit J.M. Christofides, N. Project scheduling with resource constraints: a branch and bound approach. *,European Journal of Operational Research*, 29:262–273, 1987.

[CWBG03] V. Crute, Y. Ward, S. Brown, and A. Graves. Implementing lean in aerospace-challenging the assumptions and understanding the challenges. *Technovation*, 23(12):917 – 928, 2003.

[Den91] Dennis-L. Dennis, T. *Management Science.* West Publishing, 1991.

[DM6] H. Schmeck D. Merkle, M. Middendorf. Ant colony optimization for resource-constrained project scheduling. *IEEE Transactions on Evolutionary Computation*, 2002:333–346, 6.

[EAD12] Eads registration documents. Technical report, EADS, 2012.

[EDW03] M. Vanhoucke E. Demeulemeester and W.Herroelen. A random network generator for activity-on-the-node networks. *Journal of Scheduling*, 6:13–34, 2003.

[EH80] Salah E. Elmaghraby and Willy S. Herroelen. On the measurement of complexity in activity networks. *European Journal of Operational Research*, 5(4):223 – 234, 1980.

[EMMR00] Myles Walton Earll M. Murman and Eric Rebentisch. Challenges in the better, faster, cheaper era of aeronautical desing, engineering and manufacturing. *The Lean Aerospace Initiative Report Series*, RP00-02, 2000.

[Fra07] Eckart Frankenberger. Concurrent design and realization of aircraft production flow lines - process challenges and successful design methods. In *International Conference on Engineering Design, ICED ' 07*, Cité des Sciences et de l'industrie, Paris, August 2007.

[Gas02] Victoria Elena Gastelum. Application of lean manufacturing techniques for the design of the aircraft assembly line. Master's thesis, Massachusets Institute of Technology, June 2002.

[GEM84] Jeff Cox Goldratt Eliyahu M. *The Goal: A Process of Ongoing Improvement.* Great Barrington, MA, 1984.

[GLLRK77] Ronald L Graham, Eugene L Lawler, Jan Karel Lenstra, and AHG Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics. v5*, pages 287–326, 1977.

[Glo89a] F Glover. Tabu search part i. *ORSA Journal on Computing*, 1:190–206, 1989.

[Glo89b] F Glover. Tabu search part ii. *ORSA Journal on Computing*, 2:4–32, 1989.

[Har98] S. Hartmann. A competitive genetic algorithm for resource constrained project scheduling,. *Naval Research Logistics*, 45:733–750, 1998.

[HB10] Sönke Hartmann and Dirk Briskorn. A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1):1 – 14, 2010.

[HK00] Sönke Hartmann and Rainer Kolisch. Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 127(2):394 – 407, 2000.

[HKPS11] J.L. Hurink, A.L. Kok, J.J. Paulus, and J.M.J. Schutten. Time-constrained project scheduling with adjacent resources. *Computers & Operations Research*, 38(1):310 – 319, 2011. <ce:title>Project Management and Scheduling</ce:title>.

[Hol75] H. Holland. Adaptation in natural and artificial systems. *University of Michigan Press*, 1975.

[HR99] W. Herroelen and B. De Reyck. Phase transitions in project scheduling. *The Journal of the Operational Research Society*, 50(2):pp. 148–156, 1999.

[HRS+01] G. Heike, M. Ramulu, E. Sorenson, P. Shanahan, and K. Moinzadeh. Mixed model assembly alternatives for low-volume manufacturing: The case of the aerospace industry. *International Journal of Production Economics*, 72(2):103 – 120, 2001.

[JPW96] Daniel T. Jones James P. Womack. *Lean Thinking. Banish Waste and Create Wealth in Your Organisation.* Simon and Schuster, 1996.

[KALM11] Oumar Koné, Christian Artigues, Pierre López, and Marcel Mongeau. Event-based milp models for resource-constrained project scheduling problems. *Computers & Operations Research*, 38(1):3 – 13, 2011. <ce:title>Project Management and Scheduling</ce:title>.

[Kel63] J Kelley. The critical path method: Resources planning and scheduling. *Industrial Scheduling*, pages 347–365, 1963.

[KH99] Rainer Kolisch and Sönke Hartmann. Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis. In Jan Węglarz, editor, *Project Scheduling*, volume 14 of *International Series in Operations Research & Management Science*, pages 147–178. Springer US, 1999.

[KH06] Rainer Kolisch and Sönke Hartmann. Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research*, 174(1):23 – 37, 2006.

[Kle00] R. Klein. Project scheduling with time-varying resource constraints. *International Journal of Production Research*, 38 (16)(16):3937–3952, 2000.

[KN02] T. Ibaraki K. Nonobe. Formulation and tabu search algorithm for the resource constrained project scheduling problem. *Essays and Surveys in Metaheuristics*, pages 557–588, 2002.

[Kon09] Oumar Kone. *Nouvelles approches pour la resolution du probleme d'ordonnancement de projet a moyens limites*. PhD thesis, Universite Toulouse III - Paul Sabatier, Decembre 2009.

[Lea] Lean enterprise institute. http://www.lean.org.

[LP02] John A. Jr. Lawrence and Barry Pasternack. *Applied Management Science,*. John Willey, 2002.

[LP05] A Leitner Pilla. The lean journey at the boeing company. In *ASQ Worl Conference on Quality and Improvement Proceedings*, volume 59, pages 263–271, 2005.

[MCT08] Y. . Ma, G. Chen, and G. Thimm. Paradigm shift: unified and associative feature-based concurrent and collaborative engineering. *Journal of Intelligent Manufacturing*, 19(6):625–641, 2008. resumen - English; Paginas - 625-641; ProQuest ID - 200507517; Ultima actualizacion - 2012-07-24; Lugar de publicaciOn - London; Autor de la institucion corporativa - Ma, Y-s; Chen, G; Thimm, G; DOI - 1564948841; 40533391; 53474; JITM; SPVLJITM1084519128.

[MD96] A. Colorni M. Dorigo, V. Maniezzo. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics*, 26 (Part B):29–41, 1996.

[Moh84] Rolf H. Mohring. Minimizing costs of resource requirements in project networks subject to a fixed completion time. *Operations Research*, 32(1):pp. 89–120, 1984.

[MV04] D. DEBELS L.V. TAVARES M. VANHOUCKE, J. COELHO. On the morphological structure of a network. *Vlerick Leuven Gent Working Paper Series*, 9, 2004.

[NZ98] K. Neumann and J. Zimmermann. *Handbook on recent advances in project scheduling*, chapter Methods for Resource Constrained Project Scheduling with regular and non-regular objective functions and schedule-dependent time windows. Kluver Academic Publishers, 1998.

[Ohn88] Taiichi Ohno. *Toyota Production System: Beyond Large-Scale Production*. Productivity Press, Portland, OR, 1988.

[PWW69] A. Alan B. Pritsker, Lawrence J. Watters, and Philip M. Wolfe. Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science*, 16(1):pp. 93–108, 1969.

[RFL02] Shankar Sundaram Roberto F. Lu. Manufacturing process modeling of boeign 747 moving line concepts. In *Proceedings of the 2002 Winter Simulation Conference*, 2002.

[RKD95] Arno Sprecher Rainer Kolisch and Andreas Drexl. Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science*, 41(10):1693–1703, 1995.

[Rus70] A. H. Russel. Cash flows in networks. *Management Science*, pages 357–373, 1970.

[SLT09] M. Saunders, P. Lewis, and A. Thornhill. *Research methods for business students.* Prentice Hall, 2009.

[Spe99] Bowen H Spear, S. Decoding the dna of the toyota production system. *Harvard Business Review*, 96-106, 1999.

[Spr95] Rainte Drexl Andreas Sprecher, Arno Kolisch. Semi-active, active, and non-delay schedules for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 80:94–102, 1995.

[SU05] F. STORK and M. UETZ. On the generation of circuits and minimal forbidden sets. *Mathematical Programming*, 102(1):185–203, 2005.

[TFC99] L. V. Tavares, J. Antunes Ferreira, and J. Silva Coelho. The risk of delay of a project in terms of the morphology of its network. *European Journal of Operational Research*, 119(2):510 – 537, 1999.

[TG08] J.J. Paulus J.M.J. Schutten T.A. Guldemond, J.L. Hurink. Time-constrained project scheduling. *J. Scheduling*, 11:137–148, 2008.

[Tsa06] P.F. Tsai. *Tight Flow-Based Formulations for the Asymmetric Traveling Salesman Problem and Their Applications to some Scheduling Problems.* PhD thesis, Virginia Polytechnic Institute and State University, 2006.

[VV03] M.S. Quintanilla V. Valls, F. Ballestin. A hybrid genetic algorithm for the rcpsp. Technical report, Department of Statistics and Operations Research University of Valencia, 2003.

[ZJ08] Reklaitis GV Zapata JC, Hodge BM. The multimode resource constrained multi-project scheduling problem: alternative formulations. *AIChEJournal*, 54(8):2101 – 19, 2008.

Appendix A

Number of constraints for SEE-M and OEE-M formulations

Table A.1: SEE-M Number of constraints

Order of the events	
Constraints 5.3:	1
Constraints 5.4:	$E-1$
Constraints 5.5:	$E-1$
Start-End events for each task	
Constraints 5.6:	W
Constraints 5.7 :	W
Constraints 5.8 :	W
Time per task	
Constraints 5.9:	$\frac{(E-1)(E)}{2} W \sum_w (MAX_w^{op} - MIN_w^{op})$
One mode per task	
Constraints 5.10:	$\sum_w (MAX_w^{op} - MIN_w^{op})$
Constraints 5.11 :	$\sum_{wo} P_{ow}$
Precedence constraints	
Constraints 5.12:	$\|\mathrm{PRE}\|$
Maximal Time Lags	
5.13:	MTL_w
5.14:	MTL_w
5.15 :	$\|\mathrm{MTL}\|$
Non- Parallel Constraints	
Constraints 5.16 :	$\|\mathrm{NONP}\|$
Constraints 5.16 :	$\|\mathrm{NONP}\|$
Scarce Resources	
5.18:	O
5.19:	$O(E-1)$
5.20:	$O(E-1)$
5.21:	A
5.22 :	$A(E-1)$

Table A.2: OOE-M Number of constraints

Order of the events	
Constraints 5.31:	1
Constraints 5.32 :	1
Constraints 5.33:	$E' - 1$
Constraints 5.34:	W
Constraints 5.40 :	WE'
Tasks' non-preemption	
Constraints 5.37 :	WE'
Constraints 5.38 :	WE'
Time per task	
Constraints 5.39:	$\frac{(E'-1)(E')}{2}W$
One mode per task	
Constraints 5.35:	W
Constraints 5.36 :	$E'W\sum_w((MAX_w^{op} - MIN_w^{op})\sum_o P_{ow})$
Precedence constraints	
Constraints 5.41 :	|PRE|
Maximal Time Lags	
5.42:	MTL_w
5.43:	MTL_w
5.44 :	|MTL|
Non- Parallel Constraints	
Constraints 5.45 :	|NONP|
Scarce Resources	
5.46:	OE'
5.47:	AE'

Appendix B
Computational Results

B.1 OEE Computational Results

Table B.1: SEE Computational Results for Sets 1 & 2

Instance	TT	Events	Nodes	FLB	FIS	Solution t (secs)	Optimal Sol
Set1-8	11,5	7	1546	2,3	10	1,97	8
Set1-8	11,5	9	11203	1,92	9	21,09	8
Set1-8	11,5	11	105195	1,63	11	281	8
Set1-8	12	8	13245	2,17	11	25,49	8
Set1-8	12	10	86618	1,12	9	149,45	8
Set1-8	13,5	8	6061	2,28	10	10,2	8
Set1-8	15,5	7	3777	2,3	10	7	6
Set1-8	15,5	8	13203	1,75	9	17,32	6
Set1-8	15,5	10	219403	1,12	9	282,97	6
Set1-8	16	10	204733	1,12	9	273,67	6
Set2-8	31,5	7	7069	1,85	9	10,2	7
Set2-8	31,5	8	31116	1,194	9	39,76	7
Set2-8	31,5	9	119162	0,64	8	176,98	7
Set2-8	31,5	10	511955	0,29	10	1215,95	7
Set2-8	33	7	2981	1,84	7	6,65	6
Set2-8	33	8	25778	1,194	8	46,49	6
Set2-8	33	9	45205	0,64	9	116,81	6
Set2-8	33	10	405935	0,29	10	1007,87	6
Set2-8	34,75	7	1180	1,84	7	1,79	5
Set2-8	34,75	8	1912	1,19	10	3,39	5
Set2-8	34,75	9	6926	0,64	11	19,66	5
Set2-8	34,75	10	10619	0,296	9	44,69	5
Set2-8	34,75	11	28112	0,07	8	142,52	5
Set2-8	34,75	13	75825	0	10	663,27	5
Set2-8	41	7	152	1,84	7	0,83	5
Set2-8	55	7	2804	0,64	5	8,22	5
Set2-8	85	7	1277	0,64	8	4,29	5

Table B.2: SEE Computational Results for Set 3

Instance	TT	Events	Nodes	FLB	FIS	Solution t (secs)	Optimal Sol
Set3-8	13	9	169911	0,68	9	245,64	7
Set3-8	14	6	224	2,8	8	0,48	6
Set3-8	14	6	472	2,8	8	0,61	6
Set3-8	14	7	4631	1,7	8	6,35	6
Set3-8	14	8	23407	1,13	7	34,4	6
Set3-8	14	9	94129	0,69	7	121,98	6
Set3-8	14	10	506132	0,36	8	776,76	6
Set3-8	14,5	8	4785	1,13	7	12,31	5
Set3-8	15	8	2533	1,13	8	5,37	5
Set3-8	33	8	2203	1,13	6	6,46	5
Set3-9	17	8	1799	1,18	7	4,87	5
Set3-9	17	9	5214	0,73	7	27,5	5
Set3-9	17	10	20731	0,38	7	100,5	5
Set3-9	17	11	24707	0,08	9	138,72	5
Set3-10	17	8	1781	1,64	11	4,56	5
Set3-10	17	9	14574	1,07	11	46,11	5
Set3-10	17	10	17275	0,67	8	69,78	5
Set3-10	17	11	68836	0,36	7	270,86	5
Set3-11	15	9	116110	1,31	13	314,08	8
Set3-11	16	8	15309	1,93	10	29,06	7
Set3-11	16,5	9	172701	1,31	9	396,6	7
Set3-11	16,8	9	51814	1,31	13	112,46	6
Set3-11	17	8	5662	1,93	9	16,21	6
Set3-11	17	9	29692	1,31	9	66,94	6
Set3-11	17	10	183981	0,86	9	586,13	6
Set3-11	17	11	1233159	0,54	12	4133	6
Set3-11	17,9	8	4856	1,93	12	11,98	6
Set3-11	18	9	3070	1,32	9	21,12	5
Set3-11	24	9	1324	1,31	9	7,19	5
Set3-11	24	12	1324	0,33	6	116,67	5

Table B.3: SEE Computational Results for Set 4

Instance	TT	Events	Nodes	FLB	FIS	Solution t (secs)	Optimal Sol
Set4-8	12	9	22454	0,92	9	30,5	8
Set4-8	13	6	112	3,05	7	0,3	7
Set4-8	13	7	1957	2,14	8	2,2	7
Set4-8	13	8	3341	1,43	8	6,85	7
Set4-8	13	9	29076	0,92	9	36,58	7
Set4-8	13	10	127109	0,55	9	193,25	7
Set4-8	13	11	422579	0,26	9	905,13	7
Set4-8	13,8	9	21919	0,92	8	49,12	6
Set4-8	15	10	57007	0,55	8	94,22	6
Set4-9	13	8	5250	1,57	8	9,83	7
Set4-9	13	9	20828	1,04	8	38,02	7
Set4-9	13	10	162585	0,65	8	285,26	7
Set4-10	13	8	2940	1,87	9	7,38	8
Set4-10	13	9	75062	1,29	10	107,41	8
Set4-10	13	10	151881	0,86	10	269,79	8
Set4-11	13	8	16314	2,02	8	28,2	8
Set4-11	13	9	108599	1,4	9	149,5	8
Set4-11	13	10	202190	0,94	10	473,54	8

Table B.4: OOE Computational Results for Sets 1 & 2

Instance	TT	Events	Nodes	FLB	FIS	Solution t (sec)	Optimal Sol
Set1-8	11,5	7	478	2,4	10	0,67	8
Set1-8	11,5	9	1918	2,29	10	1,93	8
Set1-8	11,5	11	10688	2,22	10	12,68	8
Set1-8	12	8	1885	1,92	8	1,9	8
Set1-8	12	10	5514	1,525	8	6,27	8
Set1-8	13,5	8	1929	1,92	8	1,59	8
Set1-8	15,5	7	857	2,2	10	0,8	6
Set1-8	15,5	8	1385	1,92	6	1,33	6
Set1-8	15,5	10	11416	1,525	10	10,72	6
Set1-8	16	10	9725	1,525	10	9,75	6
Set2-8	31,5	7	3561	3,6	8	4,66	7
Set2-8	31,5	8	10998	3,5	7	10,58	7
Set2-8	31,5	9	18506	3,43	7	20,05	7
Set2-8	31,5	10	48674	3,38	7	61,11	7
Set2-8	33	7	678	3,6	6	0,83	6
Set2-8	33	8	2852	3,5	7	2,65	6
Set2-8	33	9	10683	3,43	7	10,3	6
Set2-8	33	10	7793	3,34	6	9,63	6
Set2-8	34,75	7	141	3,6	7	0,47	5
Set2-8	34,75	8	93	3,5	7	0,58	5
Set2-8	34,75	9	630	3,43	6	1,45	5
Set2-8	34,75	10	276	3,34	6	1,11	5
Set2-8	34,75	11	177	3,33	7	1,14	5
Set2-8	34,75	13	150	3,3	5	1,45	5
Set2-8	41	7	0	3,6	6	0,2	5
Set2-8	55	7	0	3,6	5	0,16	5
Set2-8	85	7	0	3,6	5	0,14	5

Table B.5: OOE Computational Results for Set 3

Instance	TT	Events	Nodes	FLB	FIS	Solution t (sec)	Optimal Sol
Set3-8	13	9	32799	1,63	8	31,08	7
Set3-8	14	5	72	3,8	7	0,22	6
Set3-8	14	6	837	2,58	7	0,87	6
Set3-8	14	6	837	2,58	7	0,87	6
Set3-8	14	7	2207	2,14	7	2,14	6
Set3-8	14	8	8593	1,85	7	8,24	6
Set3-8	14	9	21865	1,64	7	19,31	6
Set3-8	14	10	261475	1,47	8	181,07	6
Set3-8	14,5	8	583	1,86	6	1,58	5
Set3-8	15	8	193	1,86	6	0,98	5
Set3-8	33	8	0	1,86	5	0,84	5
Set3-9	17	7	40	2,3	7	0,39	5
Set3-9	17	8	110	2	6	0,83	5
Set3-9	17	9	0	1,78	5	1,2	5
Set3-9	17	10	154	1,6	6	1,67	5
Set3-9	17	11	1128	1,45	11	3,95	5
Set3-10	17	8	3087	2,25	7	3,93	5
Set3-10	17	9	3852	2	8	7,86	5
Set3-10	17	10	520	1,8	9	2,96	5
Set3-10	17	11	977	1,63	7	3,4	5
Set3-11	15	9	52967	2,22	12	72,62	8
Set3-11	16	8	9072	2,5	9	14,02	7
Set3-11	16,5	9	91839	2,22	10	104,27	7
Set3-11	16,8	9	22988	2,22	10	27,67	6
Set3-11	17	7	793	2,89	7	1,45	6
Set3-11	17	8	4081	2,5	9	5,16	6
Set3-11	17	9	20542	2,22	8	22,99	6
Set3-11	17	10	82736	2	8	82,21	6
Set3-11	17	11	467471	1,82	7	452,53	6

Table B.6: OOE Computational Results for Set 3

Instance	TT	Events	Nodes	FLB	FIS	Solution t (sec)	Optimal Sol
Set4-8	13	5	0	4,74	7	0,11	7
Set4-8	13	6	408	2,5	9	0,65	7
Set4-8	13	7	1819	2,14	8	2,06	7
Set4-8	13	8	3460	1,88	7	3,82	7
Set4-8	13	9	16935	1,67	9	17,18	7
Set4-8	13	10	53020	1,5	8	45,66	7
Set4-8	13	11	25773	1,36	8	29,28	7
Set4-8	13,8	9	5386	1,67	7	6,22	6
Set4-8	15	10	8232	1,5	6	9,94	6
Set4-9	13	8	7490	2	8	7,83	7
Set4-9	13	9	9479	1,78	8	11,18	7
Set4-9	13	10	46245	1,6	8	47,58	7
Set4-10	13	8	20033	2,25	9	18,55	8
Set4-10	13	9	39242	2	8	46,15	8
Set4-10	13	10	101903	1,8	8	112,63	8
Set4-11	13	8	13620	2,38	9	13,07	8
Set4-11	13	9	14976	2,11	9	16,29	8
Set4-11	13	10	71714	1,9	8	76,32	8
Set4-8	12	9	15343	1,67	34	16,77	8

Appendix C

CIO 2013 Accepted Paper: A MILP Event Based Formulation fora real-world Multimode RCSP with generalized temporal constraints

A MILP Event Based Formulation for a real-world Multimode RCSP with generalized temporal constraints.

Abstract Scheduling is becoming much more important in every industry. However, the standard RCSP usually does not cover all the characteristics of real world problems. In this work, we present an Event Based MILP formulation for a Multimode Resource Constraint Problem of direct application for some industries, as aeronautical assembly lines. Taking as a starting point of the last MILP formulations for standard RCSP, our contribution is to provide a formulation which covers the multimode case and more general temporal constraints than the ones usually referred to in the literature.

Borreguero T [1], García A [2], Ortega M [3]

Keywords: scheduling, multimode, event based formulation, temporal constraints, MILP.

1 Introduction

Over the last years, the continuous changes on every industry have forced enterprises to explore new manufacturing methods in order to comply with the OTOQOC paradigm (On time, On Quality, On Cost). Production systems based on the Toyota Production System have spread worldwide as a means of reducing

[1]Tamara Borreguero Sanchidrián (✉ tamaraborreguero@alumnos.upm.es)

Grupo IOL. Dpto. Ingeniería de Organización, Administración de Empresas y Estadística -. ETSII. UPM Madrid

[2]Álvaro García Sánchez (✉ alvaro.garcia@upm.es)
Grupo IOL. Dpto. Ingeniería de Organización, Administración de Empresas y Estadística -. ETSII. UPM Madrid.

[3]Miguel Ortega Mier (✉miguel.ortega.mier@upm.es)
Grupo IOL. Dpto. Ingeniería de Organización, Administración de Empresas y Estadística -. ETSII. UPM Madrid.

waste and optimizing manufacturing processes. The aeronautical industry, since the 1990's has been including the lean techniques into its production systems. In terms of Boeing, in a Lean production system the right resources and the right tools must be applied to achieve three key Lean principles: Takt Paced Production, One Piece Flow and Pull Production (Gastelum, 2002). Scheduling and line balancing have therefore become two main enablers for lean implementation.

Aeronautical Final Assembly Lines consist on different platforms or stations. Each platform has a fixed team of workers with different skills. The line balancing enables the distribution of the work tasks among the different platforms. Afterwards, the work tasks from each platform must be scheduled in order to complete them within the required Takt Time using the minimum number of operators. We will refer to the scheduling of the tasks as the Aeronautical Platform Scheduling Problem.

This detailed scheduling has the structure of a Resource Constrained Scheduling Problem (RCSP), which was defined by Brucker (Brucker, 1999) as the allocation of scarce resources to dependent activities over time. It is a NP Hard optimization problem and is actually one of the most intractable classical problems in practice. For both its industrial relevance and its challenging difficulty, solving the RCPSP has become a flourishing research theme. (Artigues, 2010).

There have been a wide range of studies on both heuristic and metaheuristic methods for solving the RCSP, as well as different MILP models. Recently, Koné (Koné et al, 2011) proposed the use of Event Based Formulations for the RCSP. He provided a benchmark of different models (including MILP and an heuristic) and concluded that event based formulations outperformed the previous MILP models and performed even better than the heuristic for some instances.

However, Koné's Event Based Formulations deals with the standard RCSP, which includes some assumptions that are too restrictive for many practical applications (Hartmann, 2010). Therefore, it is of great interest to improve this kind of formulations so that they can be used on industrial applications. On this work, we have developed a new Event Based Formulation that covers the characteristics of an aeronautical Assembly Platform scheduling problem. Actually, its main contribution is the allowance of multiple modes per task as well as the use of more general temporal constraints. Furthermore, the alternative objective approach, focused on minimizing the cost is more suitable for nowadays industries where the total Lead Time is usually fixed by the client Takt Time.

Section 2 provides the general classification of the Aeronautical Platform Scheduling problem. Section 3 gives an overview of existing exact formulations for the RCSP. In section 4 we propose the event based multimode formulation with additional temporal constraints. Section 5 contains the results and conclusions of tests performed on several instances and Section 6 some conclusions and future research directions.

2. The Aeronautical Platform Scheduling Problem

The RCSP is a combinatorial optimization problem, defined by a 6-tuple (V, p,E,R,B,b), where V is a set of activities, p a vector of processing times per activity, E the set of temporal constraints, R the set of resources, B the resource capacity vector and b the demand matrix (resource consumption per activity) (Koné, 2009). The objective is to identify a feasible schedule, which assigns a start /completion time (Sj / Cj) to each activity as well as a resource allocation, taking into account the temporal constraints and minimizing the total project lead time.

Until 1999, there was not a common notation for RCSP. Brucker (Brucker1999) proposed a notation based on the extension of the α| β|γ generalized scheme for the machine scheduling literature. In this notation, α refers to the *resource environment*, β to the *activity characteristics* and γ to the *objective function.* According to this notation, the Aeronautical Platform Scheduling is classified as MPSm, σ, ρ | temp | $\sum c_k$ max $r_k(S,t)$:

- α = MPSm, σ, r. This stands for a multimode resource constraint project where each activity can be processed in several alternative modes and exists a set of renewable resources available for each time period during the project execution: m being the resources, σ the units of each resource available and r the maximum number of units of the resources demanded by an activity. For our particular problem, the activities are the work tasks assigned to each platform. The renewable resources are the number of operators (each of them belonging to a particular skill) and the space on the working areas, as platforms are divided into smaller areas where a limited number of operators can work simultaneously. As well as this, each mode for an activity defines a combination of operator skills, number of operators and durations. All the operators assigned to an activity must be from the same skill and the range of possible numbers of allocated operators per tasks is independent of the chosen skill.
- β = *temp*. Among the temporal constrains, there are precedence constraints (task w' can not start until task w has been completed), non-parallel constraints (tasks w and w' cannot be in progress at the same time, but there is no precedence relation between them), and maximal time lags between tasks (task w' must start within a maximal time after w has been completed).The maximal time lag, if it exists, will be zero for all the pairs of activities. All the temporal constraints are independent from the mode in which a task is executed.
- γ=$\sum c_k$ max $r_k(S,t)$. The objective function is to minimize the resource investment. The total lead time is fixed by the assembly line Takt Time, as stated on Section 1. Therefore, the objective function is to minimize the labor cost of the assembly. The operators, once assigned to a platform stay working on it for all the Takt Time and thus minimizing the labor cost is equivalent to minimizing the maximum number of operators needed throughout the Takt Time.

3. Exact Formulations for the RCSP

Most of the research on RCSP has focused on the core single-mode problem with precedence and resource constraints. In this section, we will review MILP formulations for this core problem. They can be divided on three main groups:

- *Discrete Time Formulations*: In them, the time horizon is divided into time slots. The basic discrete time formulation was proposed by Pritsker (Pritsker, 1969). Afterwards, Christofides (Christofides, 1987) proposed the Disaggregated Discrete Time formulation (DDT) that implies a larger number of constraints but, on the other hand, is a tighter formulation and therefore its linear relaxation provides a better LB. The main drawback of discrete time formulations is the increase in the number of variables as the time horizon grows.
- *Continuous time formulation*: In this formulations, the time is represented by continuous variables: Alvarez-Valdés and Tamarit, (Alvarez-Valdés and Tamarit, 1993) studied *Forbidden Sets Formulations* which involve a high number of constraints that grows exponentially and cannot be used in practice. *Flow-Based Continuous time constraints*, described by Artigues. (Artigues et al, 2003) provide a poor relaxation, compared to discrete time formulations, although it can be preferable to them for instances involving large time scale.
- *Event Based Formulations:* Event Based Formulations for the RCSP where developed by Koné in 2009 (Koné, 2009), from a model introduced by Zapata. (Zapata , 2008). These formulations define a series of events which correspond to the start or end of the different activities. They are based on the fact that for the RCSP it always exists an optimal semi-active schedule in which the start time of an activity is either 0 or coincides with the completion time of another activity (Sprecher, 1995). Therefore, at most $n+1$ events have to be considered. They have the advantage of not depending on the time horizon, making them especially relevant for long time projects, as is the case.
  Among Event Based Formulations, the Start/End Event Based Formulation involves two types of binary variables, x_{we} and y_{we} , that are equal to 1 if task w starts (in the case of x_{we}) or ends (y_{we}) at event e and 0 otherwise.

4. Model Formulation

The Start/End Event Based Formulation (SEE) has been used as a starting point for an extended formulation that copes with the multimode problem and the additional maximal separating time and non-parallel constraints as explained on Section 2. The resulting formulation uses four sets: O stands for the operator profiles, W for the work Tasks, A for the Working Areas and E for the events. The model parameters are defined in Table 2.

Due to the new characteristics of our model, we replace the original SEE variables x_{we} and y_{we} with variables x_{weon} and y_{weon} , to be set to 1 if task w starts or ends on event e, using n operators of profile o. As well as this, we define the non-negative variables r^*_{oe} to represent the amount of resource o needed immediately after event e and non-negative variables s^*_{ae} to represent the number of operators working on area a immediately after event e. A binary variable $\alpha_{ww'}$ is defined for tasks with non-parallel constraints $(w, w' \in W \; and \; NONP_{ww'} = 1 \; and \; w < w')$ set to 1 if w ends before w' starts and 0 vice versa. A continuous variable $t^i_w \geq 0$ defines the starting time of a task. This will be used for maximal time lag constraints, and defined $\forall w, w' \in W \; and \; \sum_{w'} MTL_{ww'} + MTL_{w'w} > 0$.

The continuous non-negative variable $t_e$represents the time of event e, and the free variable num^{op}_o is used for the total number of operators of profile o needed.

Table 2 Parameters

Parameter	Definition
D_w	Total amount of working hours for task $w \in W$, if assigned only to one operator
G_{nw}	Reduction coefficient to the work task w's makespan when it is done by n operators, $w \in W, MIN^{op}_w < n < MAX^{op}_w$
P_{ow}	1 if task $w \in W$ can be done by operators with profile $o \in O$, 0 otherwise
MAX^{op}_w	Maximum number of operators that can work on task w
MIN^{op}_w	Minimum number of operators that can work on task w
$PRE_{ww'}$	1 if the precedence graph includes a precedence relationship between work tasks w and w', w,$w' \in W$
$NONP_{ww'}$	1 if the precedence graph includes a non-parallel constraint between work tasks w and w': w,$w' \in W$ and $w < w'$.
$MTL_{ww'}$	1 if the precedence graph includes a maximal time lag constraint between w' and w, w,$w' \in W$ $(MTL_{WW'} \leq PREC_{WW'})$
AR_{aw}	1 if work task w is done on area a, 0 otherwise
CAP_a	Maximum number of operators that can work on area a, $a \in A$
LT	Lead time
M	Big enough number

The formulation can be written as follows (domain restrictions omitted):

$$\text{min.} \left(\sum_o num^{op}_o\right) \tag{1}$$

Subject to:

$$t_0 = 0 \tag{2}$$

$$t_{e+1} - t_e \geq 0 \ \forall \ e \ \in E \ <> last(e) \tag{3}$$

$$t_e \leq LT \ \forall \ e \ \in E \tag{4}$$

$$\sum_{eon} ey_{weon} - \sum_{eon} ex_{weon} \ \geq 1 \ \forall \ w \ \in W \tag{5}$$

$$\sum_{eon} x_{weon} = \ 1 \ \forall \ w \ \in W \tag{6}$$

$$\sum_{eon} y_{weon} = 1 \ \forall \ w \ \in W \tag{7}$$

$$t_f - t_e - \sum_{won} D_w \, G_{nw} x_{weon} + D_w G_{nw} \left(1 - \sum_{won} y_{wfon}\right) \geq 0 \ \ \forall \ f > e, \\ w \in W, MIN_w^{op} \leq n \leq MAX_w^{op} \tag{8}$$

$$\sum_{eo} x_{weon} = \ \sum_{eo} y_{weon} \ \ \ \forall \ w \ \in W, \ MIN_w^{op} \ \leq n \ \leq \ \ MAX_w^{op} \tag{9}$$

$$\sum_{en} x_{weon} = \ \sum_{en} y_{weon} \ \ \ \forall \ w \ \in W, o \ \in O \ / \ P_{Ow} = 1 \tag{10}$$

$$\sum_{\substack{e''=0 \\ on}}^{e-1} x_{we''on} + \ \sum_{\substack{e'=e \\ on}}^{E} y_{we'on} \leq 1 \ \forall \ \mathrm{w}, \mathrm{w}' \ / \ PRE_{ww'} = 1, \ \forall \mathrm{e} \ \in \mathrm{E} \tag{11}$$

$$t_w^i \ \geq t_e - \ M \ (1 - \sum_{on} x_{weon}) \ \forall e \ \in E, \forall w \ / \ \sum_{w'} MTL_{ww'} + MTL_{w'w} > 0 \tag{12}$$

$$t_w^i \ \geq + M \ (1 - \sum_{on} x_{weon}) \ \ \forall e \ \in E, \forall w \ \ / \ \sum_{w'} MTL_{ww'} + MTL_{w'w} > 0 \tag{13}$$

$$t_{w'}^i - t_w^i - \ \sum_{eon} G_{nw} DUR_w x_{weon} \ \leq 0, \ \forall w, w' \ / \ MTL_{ww'} = 1 \tag{14}$$

$$\sum_{eon} ey_{weon} - \ \sum_{eon} ex_{w'eon} \ \leq M(1 - \alpha_{ww'}) \ \forall \ w, w' / \ NONP_{ww'} = 1 \tag{15}$$

$$r_{o0}^* - \sum_{wn} nx_{w0on} = 0 \ \forall \ o \ \epsilon \ O \tag{16}$$

$$r_{oe}^* - r_{oe-1}^* + \sum_{\substack{w \\ P_{ow=1}}} \ (\sum_n ny_{weon} - \sum_n nx_{weon}) = 0 \ \forall \ o \ \ \in O, e \ \in E \tag{17}$$

$$r_{oe}^* \leq \ num_o^{op} \ \forall o \in O, e \in E \tag{18}$$

$$s_{a0}^* - \sum_{won} nx_{w0on} \, AR_{aw} = 0 \ \forall \ a \ \epsilon \ A \tag{19}$$

$$s_{ae}^* - s_{ae-1}^* + \ \sum_w (\sum_{on} ny_{weon} AR_{aw} - \sum_{on} nx_{weon} AR_{aw}) = 0 \ \ \ \forall \ a \ \in A, \ e \ \in E \tag{20}$$

$$s_{ae}^* \leq \ CAP_a \ \ \forall a \in A, e \in E \tag{21}$$

Equation (1) is the objective function: to minimize the total number of operators. Constraint (2) forces the first event to begin at t=0 and constraint (4) assures that there is no delay in the task completion. The order of the events on time is imposed by constraint (3). Constraint (5) states that the start event of a task must precede its end event. Constraints (6) and (7) limit to one the start and end per work task. Constraint (8) fixes the minimum time difference between the start and the end events to the duration of the task. A single mode for performing the task is imposed by constraints (9) and (10). As for the relations between tasks: (11) is the multimode expression for the precedence constraints. Maximal time lags equal to zero for consecutive events are expressed on constraints (12) to (14). Non-parallel constraints are (15). Resource needs are expressed on equations (16) to (18) in the case of operators and (19) to (21) for the available capacity per area.

5. Results

The computational results were obtained using CPLEX12.4 solver. The tests were carried out on an Intel-Core i7-2630QM processor with 2GHz and 4 GB RAM, running Windows 7. As the standard PSPLIB instances are not valid for the structure of the problem, four different sets of 8 task instances were used. Table 4 shows the main instance characteristics. Sets 3 and 4 were also extended in order to create instances of up to 11 tasks.

All instances were solved up to optimality, taking times from seconds to fifteen minutes. The solution time grew exponentially with the number of events, even when solving the same set of instances, see Figure 1. Defining fewer events has also a high impact on the first LP bound, which is tighter. There is always a optimal solution with no more than a number of events equal to *Card(W)+1*. However, all the tested instances had an optimal solution with fewer events than that minimum number.

For each of the eight-task instances, different lead times were tested. On average, the solution time also grew as the objective Lead Time approached to the Critical Path Lead Time, see Table 3. Most of the instances require more solution time with the same number of events when new tasks are added. Withal, some were solved faster with more tasks. This shows that in some cases the structure of the problem is more important than the number of tasks itself. The detailed instances and computational results are available on (APSP).

Table 4 Instance Characteristics

Set	Precedences	∑MTL	∑NONPL	Op. Profiles	Areas	No.Modes
Set1	6	1	1	2	2	12
Set2	8	1	1	2	2	16
Set3	7	1	1	2	2	17
Set4	7	1	1	2	2	16

Table 3 Sample solving time for different Lead Times

Instance	LT=31.5	LT=33	LT=34.75	LT=41
Set2_8 Tasks	10.2s	6,65s	1,79s	0.83s

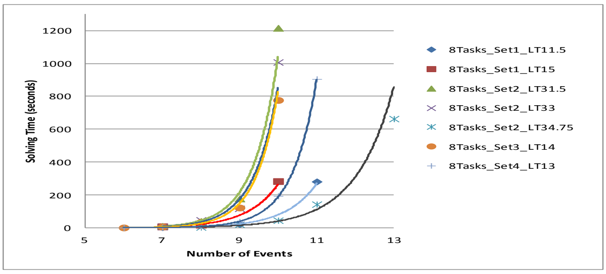

Fig. 1 Solution Time per instance with different number of events

6. Conclusions

This work provides a new MILP formulation for a real case of MRCSP. It is a first insight on the problem and has helped us identify directions for a further research. In order to extend it to bigger instances it is necessary to make a focus on the use of pre-processing to calculate the needed number of events.

As well as this, the solution times have been different for each set of instances, although they had the same task dimension. A characterization of Aeronautical Scheduling Problem instances is required in order to improve the formulation and develop pre-processing techniques suitable for both the formulation and the data sets.

7. References

Brucker P. and Knust S. (2011). Complex Scheduling. GOR-Publications. Springer

Gastelum V.E (2002) Application of Lean Manufacturing Techniques for the Design of the Aircraft Assembly Line, Massachusets Institute of Technology. Master Thesis.

APSP: https://www.dropbox.com/sh/9070da1dz6typ2o/V0jCAtR_8y

O. Kone, C. Artigues, P. Lopez, and M. Mongeau.(2011) Event-based MILP models for resource-constrained project scheduling problems. Computers and Operations Research, 38(1):3 – 13

For further references see: https://www.dropbox.com/sh/9070da1dz6typ2o/V0jCAtR_8y

The CIO-ICIEOM/2013 Program Chairs-------------

Review from Reviewer 1 -------------

Empirical basis and potential applicability : 9
Clear identification of objectives, scientific questions or hypothesis proposed in the paper : 9
Theoretical basis : 10
Research methodology : 9
Presentation and discussion of the achieved results : 8
Scientific and/or practical novelty of the paper : 9
Adjustment to the selected topic : 10
{CRITERIONNAME_08}: {CRITERIONVALUE_08}
{CRITERIONNAME_09}: {CRITERIONVALUE_09}
{CRITERIONNAME_10}: {CRITERIONVALUE_10}

-- Comments to the author(s):
Se recomienda simplemente la mejora de un pequeño aspecto formal: justificar las restricciones y la función objetivo del modelo a la izquierda en lugar de centradas, e incorporar restricciones de signo para las variables.
-- Summary:
Se recomienda simplemente la mejora de un pequeño aspecto formal: justificar las restricciones y la función objetivo del modelo a la izquierda en lugar de centradas, e incorporar restricciones de signo para las variables.
---------- End of Review from Reviewer 1 ----------------------

Review from Reviewer 2 -------------
Empirical basis and potential applicability : 9
Clear identification of objectives, scientific questions or hypothesis proposed in the paper : 9
Theoretical basis : 9
Research methodology : 10
Presentation and discussion of the achieved results : 9
Scientific and/or practical novelty of the paper : 10
Adjustment to the selected topic : 9
{CRITERIONNAME_08}: {CRITERIONVALUE_08}
{CRITERIONNAME_09}: {CRITERIONVALUE_09}
{CRITERIONNAME_10}: {CRITERIONVALUE_10}

-- Comments to the author(s):
none
-- Summary:
none
---------- End of Review from Reviewer 2 ----------
CIO-ICIEOM/2013