



A Computer Application for Scheduling in MS Project

Anabela Tereso, André Guedes and António Cascais

Department of Production and Systems, University of Minho, Guimarães 4800-058, Portugal

Corresponding author: Anabela Tereso (anabelat@dps.uminho.pt)

Abstract: The project management field has shown great progress over the last decades. With technology evolution, project managers as well as other managers have a faster and more efficient way to handle information. While progress brought more abilities to managers, it also brought them more requirements, and an increasingly higher level of minimum accepted quality. The development and usage of new scheduling techniques became therefore imperative, so that better results could be achieved. In this research, four scheduling techniques well documented in the literature were studied: Early Start Schedule, Late Start Schedule, Constructive Heuristics and Branch-and-Bound. The main objective of this research project was to integrate these scheduling techniques into commercially available software, in order to help project managers deal with scheduling tasks in a more easy and controlled way. These scheduling techniques were integrated as an add-in, coded with C# programming language, for Microsoft Project 2010. After developing the add-in, an experimental phase was performed, in which the software was tested using some example projects. The initial hypothesis was confirmed by the results. For the tested projects the conclusion was that it was possible to get better results, concerning the project's duration, using the studied techniques rather than the default scheduling technique used by Microsoft Project 2010.

Key words: Project management, project scheduling, Microsoft Project Add-in.

1. Introduction

The main objective of this research project was to integrate scheduling techniques, documented in the literature, with Microsoft Project 2010, in order to help project managers' deal with scheduling tasks in an easier and controlled way, and at the same time achieve better results.

As far as we know, there is no information about the techniques or algorithms used by Microsoft Project to do the scheduling tasks so the possibility to select the algorithms to use gives a higher degree of control to the project manager.

The initial hypothesis was that it is possible to get better results, concerning the project's duration, using the studied techniques rather than the default scheduling technique used by Microsoft Project 2010, with an increased control.

A project is, actually, not something new. Since the beginning of humanity that Mankind organizes himself

to accomplish certain tasks. Although this was made in a primitive way, and with little detail, these tasks were designed and planned before taking action, so that success could be more certain.

Over time, it was realized that there was a need to innovate and create new techniques and methodologies to manage these tasks and all the information about them. Project management emerged so that the manager could have the knowledge and the best practices to ensure that his goals are achieved within a set of established parameters, whether of time, budget or resources.

In the literatures, there are many definitions for a project. However, it can be stated that all boil down to a few key points. Most researchers say that a project is a unique and temporary activity with a well-defined goal or set of goals.

A project is defined by the PMI (Project Management Institute) [1] as a temporary endeavor to

create a product, a service or a unique result. It goes even further and claims that its temporary nature indicates that it has a start and a well-defined end. The end of a project is reached when all the goals are met or when they cannot be achieved.

Maylor [2] defines three main characteristics on which the project concept fits: unique, temporary and focused. A project is never repeated, with all its exact details. Although some of these details can be replicated to produce the same result. There are some variables, such as time, location and resources, which always create a different instance of the problem. Temporality is easily explained by the time milestones that define a project, the start and the end instants. A project always has a goal or a set of goals, which culminates in the creation of a product or a service or producing a certain expected result, hence the concept of focus.

Usually, there are three major goals in a project: performance, cost and time. Either for the development team, or the manager or even the customer, these three goals are extremely important, but the customer's expectations should also be addressed.

The project manager has the role to bring all the parts together, from the team that develops the project, based on a set of goals, to the client that imposes his conditions and limits. He articulates the information between the stakeholders, he is responsible for the final results, and he is the one who has to manage the inevitable conflicts. In the end, he is responsible for making decisions that directly affect the project's progress.

The two major milestones of a project are, undoubtedly, its beginning and its ending. Between them, there are a set of phases, each one with specific skills and goals. PMI defines this set of phases as the lifecycle of a project [1]. This set is formed by the starting of the project (initiating phase), followed by an organization and preparation phase (planning phase). Next, the actual carrying out of the work

required to reach the goals is done (executing phase), and finally the project is closed (closing phase). Each stage of the project lifecycle requires different levels of resource usage, has different costs, different durations, and even different control methods.

It is important to bear in mind that a project does not always come to an end when the proposed objectives are achieved. It may have to be terminated due to some impossibility that may arise over time, either by external factors, whether by large deviations from the management plan, among many other possible reasons. Monitoring and controlling is important throughout all the phases of the project life cycle.

This paper begins by defining what a project is, and the typical project life cycle is explained. Section 2 concentrates on project scheduling and on the scheduling techniques used, and Sections 3 and 4 present the MS Project Add-in developed and the results obtained. Finally conclusions are drawn in Section 5.

2. Project Scheduling and Scheduling Techniques

The most supported motto, by a vast majority of authors, is that project scheduling is the most important task for the success of a project. The more detailed and clear the scheduling is, the easier will be the progress of the project. There will be greater interconnection between the players involved, with consequent maximization of the efficiency of resources utilization, minimization of costs, and higher profits.

But what is, exactly, project scheduling? Scheduling is to set the guidelines that the project team has to follow to achieve success. These guidelines indicate which tasks need to be done and which resources are needed to implement them. They help to meet the customer's expectations concerning delivery dates, cost and work accomplished. The ultimate goal of a project manager is to fulfill these expectations in an optimal way, if possible.

Project scheduling, despite being detailed and objective, can be reduced to three questions: what, how and when do things need to be done? Thus, it is imperative to know, at any moment, which tasks should be performed, by whom and with which resources. But uncertainty should not be forgotten and the plan should be adjusted. Although, it is necessary to save the plan that was agreed upon at the start of the project—the baseline plan.

The most important aspect, regarding the scheduling phase, is the actual tasks' schedule. The scheduling is a way to indicate the sequence in which the activities of a project will be implemented. From this sequence of events, the project manager will be able to make estimations regarding the time of beginning and ending of each activity. These values will be crucial, so that the goal for project completion, established with the client, can be met.

But building this sequence of activities is not a trivial task. To be able to proceed with the scheduling of the project's activities it is necessary to take into account a range of details and constraints that will directly impact on this task. Each activity is characterized by an amount of time needed for its execution (the duration), an amount of resources needed to support the implementation and a set of precedence relations with other activities. All these parameters are important when allocating an activity. It can't be scheduled to run when there are insufficient resources or when one of their predecessor activities, if any, is not yet completed. This problem is known in the literature as the RCPSP (resource constrained project scheduling problem).

A simple solution for the scheduling problem, without considering resource constraints, would be to make use of the slack of an activity to move its execution time to a moment that does not interfere with another conflicting activity, nor the total project time, whenever possible.

CPM and PERT methods do not address the resource-constrained problem. Therefore, it became

imperative to develop a set of techniques to solve this problem. The RCPSP aims to find the time instants in which the activities of a given project should begin, subject to precedence and resource constraints, in order to minimize the project duration.

Due to the precedence relationships, it is common for an activity a_j to have more than one predecessor. If these predecessors do not finish at the same time, activity a_j will have to wait until the last one finishes, before it can start its execution. This makes the activities that have ended earlier as predecessors to have slack time.

In the following subsections, the scheduling techniques used in the developed add-in will be present, assuming networks in AoN (activity-on-node) format.

2.1 Early Start Schedule

Considering an AoN network represented by the graph $G = (V, E)$, with V representing the set of vertices (nodes) and E the set of edges (arcs). Assuming a dummy start node 1 and a dummy sink node n , and that the nodes are topologically numbered such that an arc always leads from a smaller to a higher node number, EST_i represents the earliest possible start time, EFT_j the earliest possible finish time of activity j , P_j the set of immediate predecessor activities of activity j and d_j the fixed activity duration of activity j . The pseudo code to obtain the early start schedule [3] is

$$\begin{aligned} EST_1 &= EFT_1 = 0; \\ \text{For } j &= 2 \text{ to } n \text{ do} \\ EST_j &= \max\{EFT_i \mid i \in P_j\}; \\ EFT_j &= EST_j + d_j; \end{aligned}$$

Analyzing this pseudo code in more detail, the first activity represents the project start dummy activity and therefore the Earliest Start Time and the Earliest Finish Time for this activity is set to be zero. For the remaining activities, the Earliest Start Time is evaluated as the maximum of the Earliest Finish Time of all the predecessors. The Earliest Finish Time is equal to the sum of the Earliest Start Time with the

duration of the activity. This procedure is normally called the forward pass calculations.

2.2 Late Start Schedule

Now consider an upper bound T on the latest allowable project completion time of the project. With the backward pass calculations the latest allowable start (finish) time of activity i , LST_i (LFT_i) can be evaluated. S_i denotes the set of immediate successor activities of activity j . To obtain the late start schedule, the backward pass calculations are applied, after the previous procedure is applied. The pseudo code to obtain the late start schedule is

$$LST_n = LFT_n = T;$$

For $i = n - 1$ down to 1 do

$$LFT_i = \min\{LST_j \mid j \in S_i\};$$

$$LST_i = LFT_i - d_i;$$

For this procedure, the late start and finish times of the dummy end activity of the project are set equal to T . For all other activities, the latest allowable finish time of an activity is the minimal latest allowable start time of its immediate successors. Subsequently, the latest allowable start time of an activity is computed as its latest allowable finish time minus its duration.

2.3 Constructive Heuristics

In a simplistic way, it can be stated that the RCPSP consists in finding the answer to the following question: what is the best way to schedule the activities of a project, to finish it as quickly as possible, taking into account the resource constraints?

The trend of the research on this problem is experiencing major changes lately [4]. While the earliest forms of resolution were based on mathematical optimization, the trend lately is to make use of heuristic procedures, which allows reaching a solution to the problem in reasonable time, for medium and large projects, as those found in practice. Other techniques that also produce successful results are the exact algorithms (see Branch-and-Bound section), with the disadvantage of their runtime.

These heuristic procedures consist of three basic parameters, a priority rule, a scheduling scheme and a direction in time for this scheme. The priority rule indicates the order in which the activities will be scheduled; the scheduling scheme determines how the construction of a valid solution is done, sequentially assigning the respective start times for each activity scheduled; and the direction in time that indicates whether this scheduling scheme sequence is done in the Forward Direction of time, or if it is done in the Backward Direction.

2.3.1 Priority Rules

The priority rules are no more than rules to decide the order in which the activities will be added to the schedule.

There are several studies on the behavior and performance of these rules [5-7].

The priority rules used were: MINEST (Minimum Early Starting Time), MINLST (Minimum Late Starting Time), MINEFT (Minimum Early Finishing Time), MINLFT (Minimum Late Finishing Time), MINSLK (Minimum Activity Slack), GRD (Greatest Resource Demand), LPT (Largest Processing Time) and SPT (Shortest Processing Time).

2.3.2 Scheduling Schemes

- SSS (serial schedule scheme)

With the help of the serial schedule scheme it is possible to construct a schedule for a set of activities in n iterations.

To start the scheduling, the list of activities sorted, based on the priority rule chosen, is obtained.

Each iteration, the first activity of the sorted list, is selected, which is then added to a PS (Partial Schedule). That activity is allocated with the Earliest Start Time in which it could be executed, since all its predecessors are already completed and no resource conflict occurs throughout its duration. After the allocation of this activity, it is necessary to update the values of resource availability and to do the calculations of the earliest and latest starting and ending times of the remaining activities.

When it is impossible to schedule an activity in a specific time instant, because there are still ongoing predecessor activities, or because there are resource conflicts, it will be delayed until the first time instant in which such allocation is possible.

The next iteration will try to schedule another activity, following the previous procedure, and so on until the end dummy activity is included in the schedule.

- PSS (parallel schedule scheme)

The PSS works on the basis of time instants, i.e., instead of scheduling an activity in each iteration, zero or more activities are scheduled at each time instant t . Time instant t is chosen so that all activities have scheduled start before t . In each iteration, the set of eligible activities is evaluated, ordered by the priority rule previously chosen. The next step is to verify which of these activities have an early start equal or earlier than time instant t . For each activity that passed this filter, the possibility to schedule it at time t without violating any resource constraint throughout its execution is analyzed. If this is possible, it will be added to the PS. The precedence constraints in this case are already assured at the time of calculating the number of eligible activities. If a chosen activity cannot be scheduled at that time instant, it is discarded and removed from the activities ready to be scheduled, and the next one will be chosen.

At the end of the iteration, a new value will be computed for t , which will be the smallest time instant termination of all activities of the PS. For this time instant, the set of eligible activities will also be updated, including all activities that failed to be scheduled in the previous iteration cycle.

2.4 Branch-and-Bound

The Branch-and-Bound procedure belongs to the family of exact methods, and can be used for solving the RCPS. This procedure divides the main problem into smaller problems, along a search tree, where the nodes represent those sub problems.

This work analyzed the Branch-and-Bound procedure developed by Demeulemeester and Herroelen [3], also known in the literature by DH-procedure, which was made available in the add-in developed.

Partial schedules have their starting point at time instant zero. Thereafter, at each decision point m , subsets of activities are added (the empty set \emptyset is also possible), until a valid schedule is reached, considering all restrictions. Thus it can be stated that a complete schedule is an extension of a partial schedule.

At each decision point m , the eligible activities E_m are evaluated.

The Branch-and-Bound algorithm used is described in the book by Demeulemeester and Herroelen [3] in pages 359-362.

3. MS Project Add-in

The MS Project Add-in was developed using the Microsoft Visual C Sharp (C#) programming language in the Integrated Development Environment Microsoft Visual Studio 2010 (VS2010) on the Windows 7 operating system. Some features of the Microsoft Visual Basic (VB) programming language were also used, because they were not available in C#.

With the help of libraries to integrate Microsoft Office solutions in VS2010, specially the Microsoft.Office.Interop.MSProject library, it was possible to interact with the MS Project application and manipulate the inserted data.

Another library that was also very helpful was the Microsoft.Office.Tools.Ribbon because it gives the opportunity to change an existing tab, or create a new one, with features added by the developer. The choice made was to create a new tab (Fig. 1), with several buttons that allow the end user to choose the algorithm to schedule its project within MS Project.

As can be seen in Fig. 1, each of the four methods has a reserved area in the tab developed. However, only the Constructive Heuristics has an area for

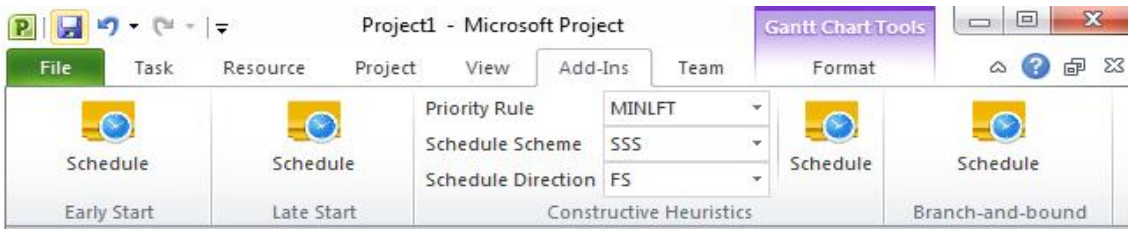


Fig. 1 Tab with the new MS Project Add-in.

parameters introduction, before the actual scheduling is done. The parameters are: the Priority Rule chosen, the Schedule Scheme and the Schedule Direction. The combination of these three parameters yields a total of thirty two possibilities, which translates into thirty two different methods of scheduling the project, using the Constructive Heuristics. After selecting the three parameters in the case of the Constructive Heuristics, it is time to click on the Schedule button to run the specific algorithm.

Throughout the development of the add-in in C#, it was necessary to create two additional classes, in addition to the class that contains all properties and methods of the add-in tab (ScheduleAdd-in). The Node class has four properties that store all the information of a node of the search tree, constructed during the execution of the Branch-and-Bound algorithm. These four properties are the search tree level, the decision point, the set of tasks belonging to a cut set and the active tasks. The second class that was created is directly dependent on the previous one. The StoredInfo class has two properties: a list of tasks belonging to the Partial Schedule and a tree node, which contains all the information belonging to an instance of the class Node. The class StoredInfo is also used during the execution of the Branch-and-Bound algorithm, and has the function of storing the information at the end of a step, which can be restored a few steps ahead. Fig. 2 represents the classes mentioned here and their interconnections.

Both in the Constructive Heuristics and in the Branch-and-Bound method, the Schedule button brings up a dialog box for the user to define the maximum amount of resources available for each type of resource per time period (Fig. 3).

After the user enters the desired value and presses the OK button, the algorithm will compute the start and end times of all activities, and will adjust the Gantt chart to represent these changes.

In addition to the treatment of the data from MS Project files, entered directly by the user, it is also possible to upload projects in the test environment from VS2010, with a simple set of operations in C#. In Fig. 4, you can see a small example of adding an activity to a project and changing its properties, such as the duration and resource usage, through a piece of code in C#.

4. Results

To allow performing tests to the developed add-in, some project examples were created. The different methods studied were tested with all these projects. Due to the extension of the results obtained and to limitations of space, only some of these projects, with the results obtained for a representative set of methods, will be presented in this paper.

So, to illustrate the application of the developed add-in, two project examples were used. The first one is a simple project network, with eight activities and one resource type, which will be used to illustrate the

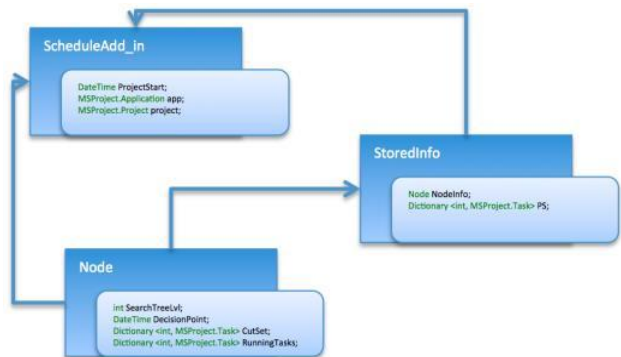


Fig. 2 Project's class diagram.

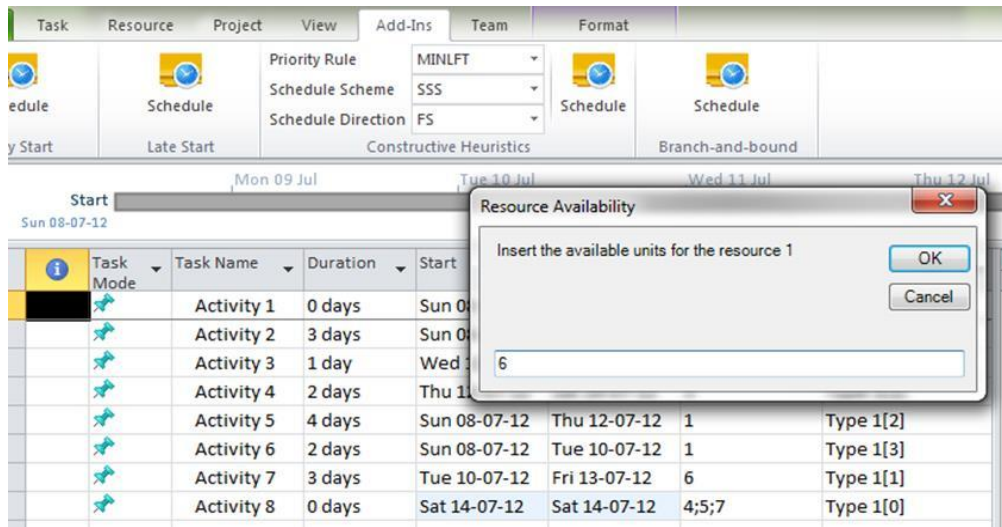


Fig. 3 Dialog box to insert the resource availability.

```

34 MSPProject.Task task2 = p.Tasks.Add("Activity 2", missing);
35 task2.Duration = "3";
36 task2.Predecessors = "1";
37 MSPProject.Assignment assignment0201 = task2.Assignments.Add(task2.ID, 1);
38 assignment0201.Units = 1;
    
```

Fig. 4 A piece of C# code to upload test projects.

application of the Early Start Schedule and Late Start Schedule techniques. The second one is a slightly more complex project network with ten activities and three resource types, which will serve to illustrate the application of the rest of the scheduling techniques.

The results presented are the Gantt charts obtained where the total project duration can be seen, for each scheduling technique implemented.

4.1 Early Start and Late Start Schedules

Table 1 shows the parameters for the example project 1 and Fig. 5 shows the associated AoN network. As these scheduling techniques do not take into account the availability of resources, it was selected a project which contains a critical and two non-critical paths, thus giving the possibility to observe the differences between the early and late start schedule. These techniques assume unlimited resource availability.

Fig. 6 represents the Early Start Schedule obtained for the example project, which is identical to schedule generated by MS Project, without resource constraints.

Fig. 7 represents the Late Start Schedule obtained for the example project, which is also identical to the schedule generated by MS Project, if the user selects to schedule activities as late as possible.

Table 1 Parameters for example project 1.

Activity	Duration	Resources	Predecessors
1	0	0	-
2	3	1	1
3	1	2	2
4	2	2	3
5	4	2	1
6	2	3	1
7	3	1	6
8	0	0	4;5;7

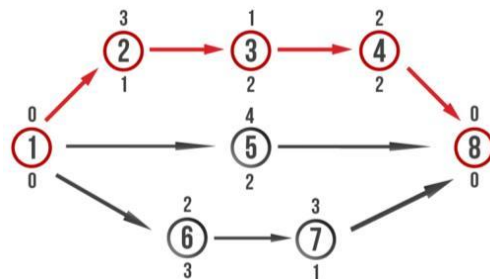


Fig. 5 AoN network for example project 1.

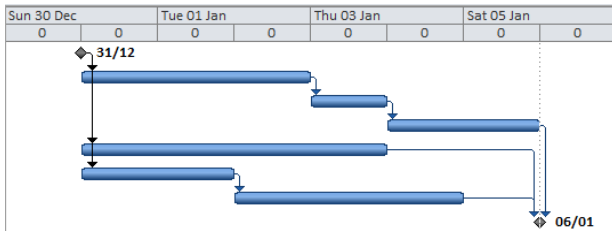


Fig. 6 Early Start Schedule (Add-in & MS Project).

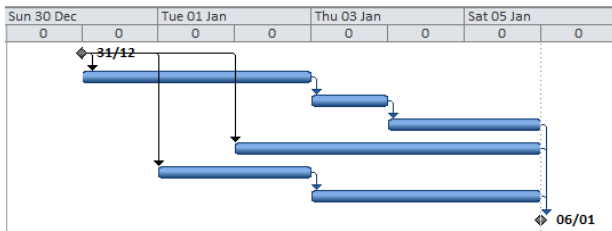


Fig. 7 Late Start Schedule (Add-in & MS Project).

4.2 Constructive Heuristics

Table 2 shows the parameters for the example project 2. Fig. 8 represents the AoN network for the same project. In the following scheduling techniques, the second project example was used, with three types of resources, in order to obtain a greater variety of results.

For this problem, the following resources availabilities were considered:

$$a_1 = 6, a_2 = 4, a_3 = 8$$

For the thirty two possible combinations to apply may be observed. Some show a reduction on the total duration of the project, some give similar results to those of MS Project (using manual resource leveling), and some show worst results.

Fig. 9 shows an example of the results obtained using one of these scheduling techniques, which represents a typical result. Fig. 10 shows the results obtained by MS Project, so a more objective comparison can be made.

4.3 Branch-and-Bound

For the Branch-and-Bound exemplification, the same project (example project 2) was used.

After computing all resource constraints and precedence relationships, the result of the application of the Branch-and-Bound technique is presented in Fig.

Table 2 Parameters for example project 2.

Activity	Duration	Resources	Predecessors
1	0	0	-
2	2	2;1;1	1
3	6	1;3;1	2
4	5	5;1;1	1
5	2	1;4;1	4
6	1	1;2;6	5
7	3	1;1;1	1
8	3	2;2;2	7
9	2	1;3;1	8
10	0	0	3;6;9

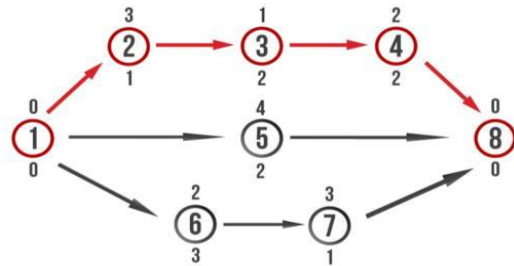


Fig. 8 AoN network for example project 2.

11. It can be seen that it represents a good alternative to the other schedules.

4.4 The PSPLIB Examples

In order to better test the developed add-in, we accessed the Project Scheduling Problem Library (PSPLIB) and downloaded some examples from the available data [8].

From the examples downloaded, we have used the instance set j3029. As we can see in Table 3, this instance set has 10 examples which were tested using 4 different algorithms: Early Start Schedule (ESS), Late Start Schedules (LSS), Constructive Heuristics (CH) and Branch-and-Bound (B&B). To apply the Constructive Heuristic, we have chosen a Serial Scheduling Scheme, with Forward Planning and the Shortest Processing Time (SPT) heuristic.

The ESS and LSS algorithms ignore the resources, so it is normal to obtain better results than the SPT heuristic, which takes twice as much time to complete all the tasks. As the B&B algorithm is an optimal one,

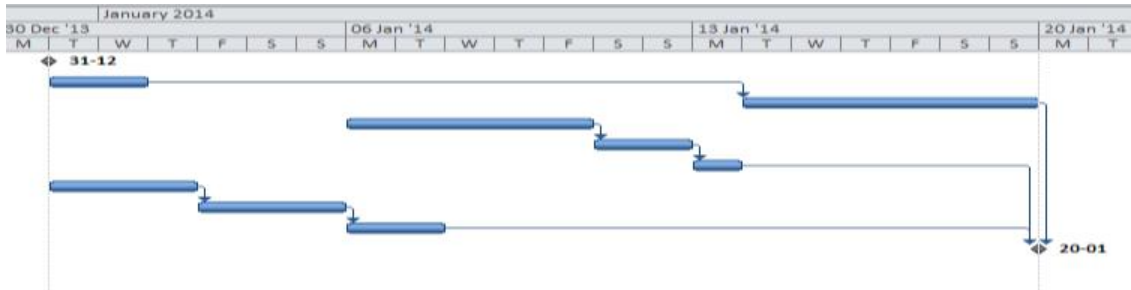


Fig. 9 Scheduling obtained using Constructive Heuristics (SPT-SSS-FS).

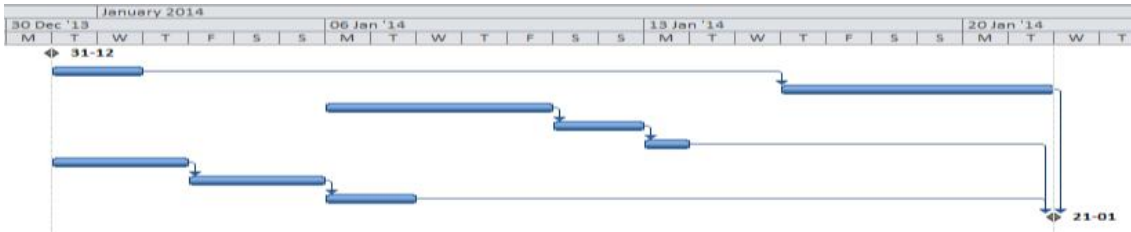


Fig. 10 Scheduling obtained using MS-Project with manual resource leveling.

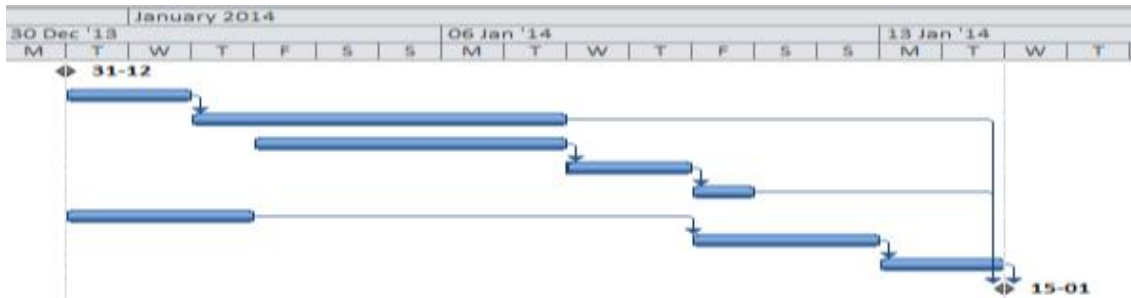


Fig. 11 Scheduling obtained using branch-and-bound.

Table 3 Project makespan obtained for the PSPLIB instances using early start and late start schedules, constructive heuristics and branch-and-bound.

Makespan	ESS	LSS	CH	B&B
J3029_1	62	62	104	85
J3029_2	55	55	126	90
J3029_3	45	45	96	78
J3029_4	54	54	120	103
J3029_5	53	53	142	98
J3029_6	43	43	107	92
J3029_7	41	41	108	73
J3029_8	50	50	112	80
J3029_9	52	52	124	97
J3029_10	54	54	96	77

it gives better results than Constructive Heuristic.

In Table 3, the results are expressed in the number of days necessary to complete the project (project makespan).

5. Conclusions and Future Research

The first conclusion that can be drawn is that it was possible to implement scheduling techniques, available in the literature, in commercial software (MS Project), in the form of an add-in, to help users on the scheduling task, making it easier and controllable. The possibility made available to the user to select the scheduling method is important, allowing more control on the results obtained and a comparison of the different solutions.

Then, to test this software application developed, tests were made, with various project examples. Branch-and-Bound was the technique that achieved better results. For project example 2, Branch-and-Bound obtained a schedule with duration equal to 15 days, while MS Project reached the minimum of 21 days.

In projects with one or two different types of resources, and few critical paths, the results do not show a great variation between methods. This is due to the low complexity of the project networks, which results in schedules that do not differ much. Results show significant differences for projects that have a large number of activities. For these projects, the schedule computed by MS Project is dominated by almost all the methods used.

The best results belong to the Branch-and-Bound method and Constructive Heuristics, composed by Parallel Scheduling Scheme, the priority rule MINLFT and the Backward Direction. The worst outcome is for the heuristic using the Serial Scheduling Scheme, the priority rule GRD and the Forward Direction.

As the Branch-and-Bound method is an implicit enumeration technique, all possible solutions to the problem were analyzed, and it was chosen the one having a shorter duration, since this is the variable to be minimized.

Thus, the conclusion was that the resource leveling method used by MS Project to schedule a project under resource constraints can be used for simple situations, where there are a small number of activities and few different types of resources. Even in these cases, its performance is poor and it is not automatic.

When it comes to larger and more complex networks, with regard to the variety of resources available, the technique used by MS Project is overtaken by the majority of the methods studied, mainly by Branch-and-Bound and the Constructive Heuristics using the PSS, MINLFT and Backward Direction.

The software developed could still be improved in terms of user interface. There are a number of other approaches to the RCPSP that should also be interesting to address in the future, namely the use of meta-heuristics, like the pseudo particle swarm optimization, recently proposed to solve this problem [9].

References

- [1] PMI, A Guide to Project Management Body of Knowledge: PMBOK® Guide, Project Management Institute, 2005.
- [2] H. Maylor, Project Management, Financial Times Prentice Hall, 2005.
- [3] E. Demeulemeester, W. Herroelen, Project Scheduling: A Research Handbook, International Series in Operations Research & Management, Vol. 49, 2002.
- [4] P. Brucker, A. Drexl, R. Möhring, K. Neumann, E. Pesch, Resource-constrained project scheduling: Notation, classification, models, and methods, *European Journal of Operational Research* 112 (1) (1999) 3-41.
- [5] G. Ulusoy, L. Özdamar, Heuristic performance and network/resource characteristics in resource-constrained project scheduling, *Journal of the Operational Research Society* 40 (12) (1989) 1145-1152.
- [6] E.W. Davis, J.H. Patterson, A comparison of heuristic and optimum solutions in resource-constrained project scheduling, *Management Science* 21 (8) (1975) 944-955.
- [7] F.F. Boctor, Heuristics for scheduling projects with resource restrictions and several resource-duration modes, *The International Journal of Production Research* 31 (11) (1993) 2547-2558.
- [8] PSPLIB, Project Scheduling Problem Library: Single Mode Data Sets [Online], <http://www.omdb.wi.tum.de/psplib/files/j30.sm.zip>.
- [9] M.M. Nasiri, A pseudo particle swarm optimization for the RCPSP, *The International Journal of Advanced Manufacturing Technology* 65 (5-8) (2012) 1-10.