Febbraio 2009 *RR*-03.09

Lucio Bianco, Massimiliano Caramia

Minimizing the Completion Time of a Project Under Resource Constraints and Feeding Precedence Relations: a Lagrangian Relaxation Based Lower Bound

Minimizing the Completion Time of a Project Under Resource Constraints and Feeding Precedence Relations: a Lagrangian Relaxation Based Lower Bound

Lucio Bianco * Massimiliano Caramia †

Abstract

In this paper we study an extension of the classical Resource-Constrained Project Scheduling Problem (RCPSP) with minimum makespan objective by introducing a further type of precedence constraints denoted as "Feeding Precedences" (FP). This kind of problem happens in that production planning environment, like make-to-order manufacturing, when the effort associated with the execution of an activity is not univocally related to its duration percentage and the traditional finish-to-start precedence constraints or the generalized precedence relations cannot completely represent the overlapping among activities. In this context we need to introduce in the RCPSP the FP constraints. For this problem we propose a new mathematical formulation and define a lower bound based on a resource constraints Lagrangian relaxation. A computational experimentation on randomly generated instances of sizes of up to 100 activities show a better performance of this lower bound with respect to others. Moreover, for the optimally solved instances, its value is very close to the optimal one.

Keywords: Feeding precedences, Generalized precedence relations, Lagrangian relaxation, Makespan, Production planning.

1 Introduction

Production planning is a relevant problem in industrial processes. It consists in defining the quantities to produce for each product and the times at which production processes associated to each product have to start. In doing this, the production planner has to take

*Dipartimento di Ingegneria dell'Impresa, Università di Roma "Tor Vergata", Via del Politecnico, 1 - 00133 Roma, Italy. e-mail: bianco@disp.uniroma2.it

†Dipartimento di Ingegneria dell'Impresa, Università di Roma "Tor Vergata", Via del Politecnico, 1 - 00133 Roma, Italy. e-mail: caramia@disp.uniroma2.it

into account constraints related to the resource availability, the demand satisfaction, and the temporal relations among activities.

Temporal constraints are used to regulate e.g. separations between the starting/finishing time of a certain activity (process) and the starting/finishing time of a successor activity (process). An easy situation is that offered by the Finish-to-Start relations with zero time lag, i.e., those constraints in which it suffices to constraint the starting time of an activity to be greater than or equal to the finishing time of an immediate predecessor activity. More complex situations are those in which the decision maker has to model systems in which overlapping among processes is allowed (construction industry). These latter situations ask for the more complex Generalized Precedence Relationships (GPRs), see, e.g., Elmaghraby and Kamburoski (1992), Bartush et al. (1988), De Reyck (1998), and Bianco and Caramia (2007).

GPRs allow one to model minimum and maximum time-lags between a pair of activities (see, e.g., Demeulemeester and Herroelen, 2002, Dorndorf, 2002, and Neumann et al., 2002). A time lag is an amount of time that must elapse at least (minimum time-lag) or at most (maximum time-lag) between the starting/finishing time of an activity and the starting/finishing time of another activity. For the sake of completeness, we recall that four types of GPRs can be distinguished: Start-to-Start (SS), Start-to-Finish (SF), Finish-to-Start (FS) and Finish-to-Finish (FF).

A minimum time-lag ($SS_{ij}^{\min}(\delta), SF_{ij}^{\min}(\delta), FS_{ij}^{\min}(\delta), FF_{ij}^{\min}(\delta)$) specifies that activity j can start (finish) only if its predecessor i has started (finished) at least δ time units before.

Analogously, a maximum time-lag ($SS_{ij}^{\max}(\delta), SF_{ij}^{\max}(\delta), FS_{ij}^{\max}(\delta), FF_{ij}^{\max}(\delta)$) imposes that activity j should be started (finished) at most δ time slots beyond the starting (finishing) time of activity i.

In order to better understand GPRs we report some examples. If, for instance, a company must place a pipe (activity j) in a given region, it is necessary to prepare the ground (activity i) in advance. This situation can be represented by a constraint $SS_{ij}^{\min}(\delta)$, since the start of activity j must be δ units of time forward the starting time of activity i. In another project, if a company must supply a client with a certain number of products which must be also assembled within 100 days, this relationship can be modelled as $SF_{ij}^{\max}(100)$, which says that the assembly process (activity j) must finish at most 100 days after the starting time of the delivery (activity i) of the products.

However, sometimes also GPRs are not in charge to fully describe the planning problem under consideration. This happens for production planning in make-to-order manufacturing companies which commonly requires the so-called project-oriented approach. In this approach a project consists of tasks, each one representing a manufacturing process, that is an aggregate activity. Due to the physical characteristics of these processes the effort associated with a certain activity for its execution can vary over time. An example is that of the human resources that can be shared among different simultaneous activities in proportion variable over time. In this case the amount of work per time unit devoted to each activity, so as its duration, are not univocally defined.

This kind of problems is in general modelled by means of the so called Variable Intensity formulation, that is a variant of the Resource Constrained Project Scheduling Problem (see, e.g., Kis, 2006). In this formulation a variable intensity is introduced for each activity to define the effort spent to process the activity in each time period. In the context defined before the actual production is such that an activity usually starts at low intensity and then gradually increases to a maximum. The resources needed to complete an activity are consumed proportionally to the varying of intensity.

It follows that the durations of activities cannot be taken into play, generalized precedence relations cannot be used any longer, and we need to introduce the so called "feeding precedences" (see, e.g., Kis et al. 2004, Kis, 2005, 2006, and Alfieri et al., 2008). Feeding precedences are of four types:

- *Start-to-%Completed between two activities* (i, j). This constraints imposes that the processed percentage of activity j successor of activity i can be grater than $0 \leq g_{ij} \leq 1$ only if the execution of i has already started (see Figure 1).

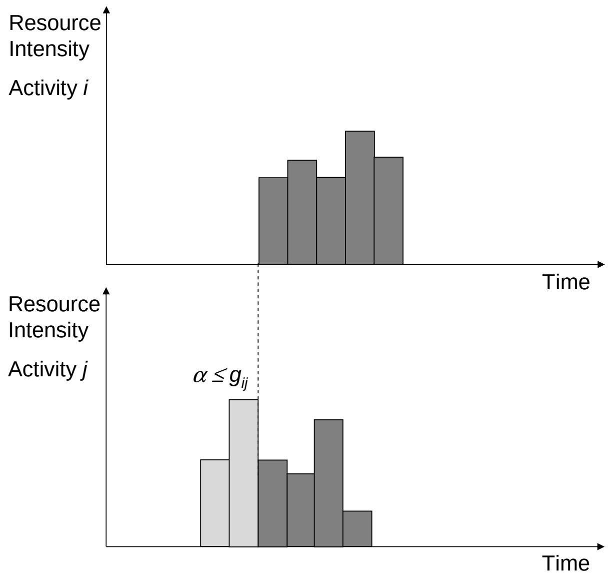

Figure 1: Example of a Start-to-%Completed constraint between activities i and j.

- *%Completed-to-Start between two activities* (i, j). This constraints is used to impose that activity j successor of activity i can be executed only if i has been processed for at least a fractional amount $0 \leq q_{ij} \leq 1$ (see Figure 2).

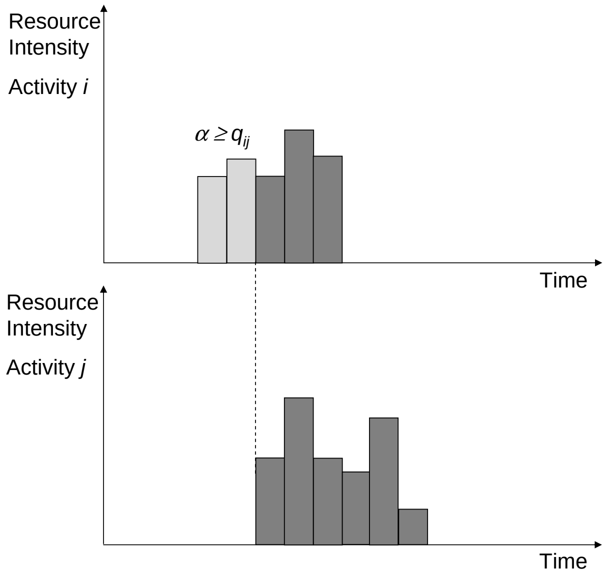

Figure 2: Example of a %Completed-to-Start constraint between activities i and j.

- *Finish-to-%Completed constraints between two activities* (i, j). This constraint imposes that the processed fraction of activity j successor of activity i can be greater than $0 \leq g_{ij} \leq 1$ only if the execution of i has been completed (see Figure 3).

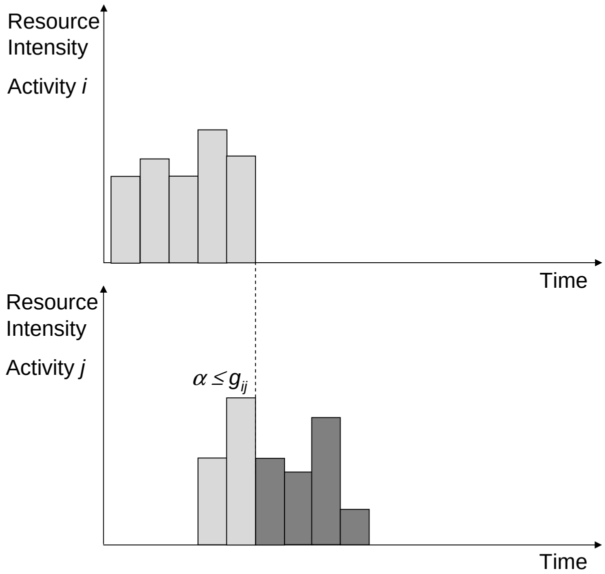

Figure 3: Example of a Finish-to-%Completed constraint between activities i and j.

- *%Completed-to-Finish constraints between two activities* (i, j). This constraint imposes that the execution of activity j successor of i can be completed only if the fraction of i processed is at least $0 \leq q_{ij} \leq 1$ (see Figure 4).

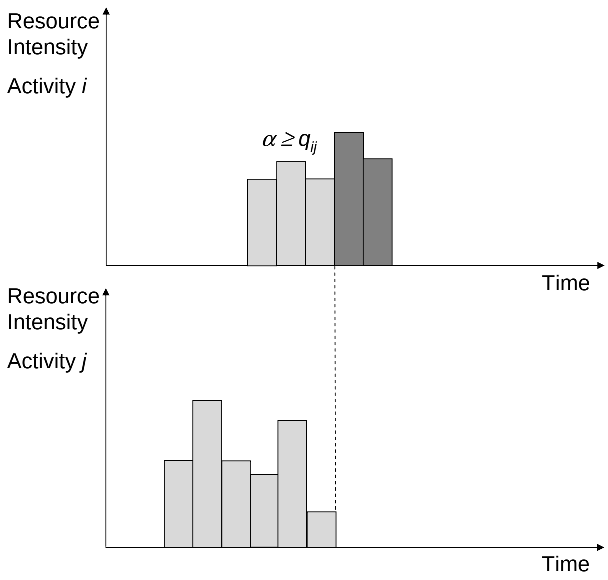

Figure 4: Example of a %Completed-to-Finish constraint between activities i and j.

Another domain of application of feeding precedence constraints could be production scheduling on different time horizons (long-term and short-term planning).

Examples of application of feeding precedence constraints to some manufacturing processes can be found in Alfieri et al. (2008).

In this paper, we study the problem of scheduling activities under feeding precedence constraints, scarce resources and minimum makespan as objective. Due to its NP-hard complexity, this problem is optimally solvable for a very limited number of activities ($20 \div 25$) within an acceptable computing time. For projects with a greater number of activities the exact duration cannot be known a-priori and therefore at least an estimate is necessary. To this end, in the following, we propose a lower bound based on a resource constraints Lagrangian relaxation.

Section 2 describes a mathematical formulation of the feeding precedence constraints in terms of mixed integer programming. Section 3 describes the problem with the objective of minimizing the completion time, showing that it is polynomially solvable. In Section 4 we give the formulation of the problem with limited resources and describe the Lagrangian relaxation along with its resolution. Section 5 is devoted to the experimental analysis of the proposed lower bound and a comparison with the optimal solution and other lower bounds obtained by different kinds of relaxation.

2 A Mathematical Formulation of Feeding Precedence Constraints

In the following, we will assume that the planning horizon within which all the production processes have to be scheduled is $[0, T)$, where T is the project deadline, and it is discretized (without loss of generality) into T unit-width time periods $[0, 1), [1, 2), \ldots, [T - 1, T)$. Let us define with

- q^1_{ij}, the fraction of activity i that has to be at least completed in order to let activity j start;
- q^2_{ij}, the fraction of activity i that has to be at least completed in order to let activity j finish;
- g^1_{ij}, the fraction of j that can be at most completed before the starting time of activity i;
- g^2_{ij}, the fraction of j that can be at most completed before the finishing time of activity i;
- A is the set of activities to be carried out;
- A_1 the set of pairs of activity for which a Start-to-%Completed constraint exists;
- A_2 the set of pairs of activity for which a %Completed-to-Start constraint exists;
- A_3 the set of pairs of activity for which a Finish-to-%Completed constraint exists;
- A_4 the set of pairs of activity for which a %Completed-to-Finish constraint exists.

Furthermore, let us consider the following decision variables

- x_{it}, the percentage of i executed till time period t.
- s_{it}, a binary variable that assumes value 1 if activity i has started in a time period $\tau \leq t$, and assumes value 0 otherwise.
- f_{it}, a binary variable that assumes value 1 if activity i has finished in a time period $\tau \leq t$, and assumes value 0 otherwise.

The feeding precedence constraints can be modelled as follows

$$
\begin{array}{lll}
x_{jt} \le s_{i,t-1} + g^1_{ij} & \forall (i,j) \in A_1, t = 1, \ldots, T & (1) \\
s_{jt} \le x_{i,t-1} + (1 - q^1_{ij}) & \forall (i,j) \in A_2, t = 1, \ldots, T & (2) \\
x_{jt} \le f_{i,t-1} + g^2_{ij} & \forall (i,j) \in A_3, t = 1, \ldots, T & (3) \\
f_{jt} \le x_{i,t-1} + (1 - q^2_{ij}) & \forall (i,j) \in A_4, t = 1, \ldots, T & (4) \\
x_{it} \le x_{i,t+1} & \forall i \in A, t = 1, \ldots, T-1 & (5) \\
s_{it} \le s_{i,t+1} & \forall i \in A, t = 1, \ldots, T-1 & (6) \\
f_{it} \le f_{i,t+1} & \forall i \in A, t = 1, \ldots, T-1 & (7) \\
s_{iT} = 1 & \forall i \in A & (8) \\
f_{iT} = 1 & \forall i \in A & (9) \\
x_{iT} = 1 & \forall i \in A & (10) \\
s_{i0} = 0 & \forall i \in A & (11) \\
f_{i0} = 0 & \forall i \in A & (12) \\
x_{i0} = 0 & \forall i \in A & (13) \\
s_{it} \ge x_{it} & \forall i \in A, t = 1, \ldots, T & (14) \\
x_{it} \ge f_{it} & \forall i \in A, t = 1, \ldots, T & (15) \\
s_{it} \in \{0,1\} & \forall i \in A, t = 1, \ldots, T & (16) \\
f_{it} \in \{0,1\} & \forall i \in A, t = 1, \ldots, T & (17) \\
x_{it} \ge 0 & \forall i \in A, t = 1, \ldots, T & (18)
\end{array}
$$

Constraints (1) model a Start-to-%Complete feeding constraint: if $s_{i,t-1}$ is equal to zero, i.e., i has not started till time t, then the amount of j that has been processed must be less than or equal to g^1_{ij}. If $s_{i,t-1} = 1$, then the amount of j processed can be greater than g^1_{ij}.

Constraints (2) model a %Complete-to-Start feeding constraint: if the total amount of activity i processed till $t-1$ is greater than or equal to q^1_{ij} then s_{jt} is less than or equal to a quantity at least equal to one, and, therefore, s_{jt} can be either zero or one; if, instead, $x_{i,t-1}$ is less than q^1_{ij} then s_{jt} must be necessarily zero, i.e., j must start after t.

Constraints (3) and (4) are the same as constraints (1) and constraints (2), respectively, for the Finish-to-%Complete and the %Complete-to-Finish constraints.

Constraints (5) regulate the total amount processed of an activity $i \in A$ over time. Constraints (6) imply that if an activity $i \in A$ is started at time t, then variable $s_{i\tau} = 1$ for every $\tau \ge t$, and, on the contrary, if activity i is not started at time t, $s_{i\tau} = 0$ for every $\tau \le t$. Constraint (7) is the same as constraint (6) when finishing times are concerned.

Constraints (8) and (9) say that every activity $i \in A$ must start and a finish within the planning horizon, respectively. Constraints (10) impose that at time period T the execution percentage x_{it} of every activity $i \in A$ must be equal to 1.

Constraints (11), (12) and (13) represent initialization conditions for variable s_{it}, f_{it}, x_{it} when $t = 0$.

Constraints (14) force the amount processed until t for an activity $i \in A$, i.e., x_{it}, to be zero if $s_{it} = 0$.

Constraints (15) force f_{it} for an activity $i \in A$ to be zero if $x_{it} < 1$.

Constraints (16), (17), and (18) limit the range of variability of the variables.

2.1 The problem with completion time objective function and unlimited resources

Let us now examine the problem of scheduling activities under feeding precedence relationships and minimum makespan as objective function.

Since the completion time of an activity $i \in A$ can be expressed as

$$f_i = \left(T - \sum_{t=1}^{T} f_{it} + 1 \right),$$

we can write the following mixed integer programming:

$$\begin{array}{lll} \min \{\max_{i \in A} f_i\} & = \min \left\{ \max_{i \in A} \left(T - \sum_{t=1}^{T} f_{it} + 1 \right) \right\} & \\ x_{jt} \leq s_{i,t-1} + g_{ij}^1 & \forall (i,j) \in A_1, t = 1, \ldots, T & (1) \\ \ldots & \ldots & \\ \ldots & \ldots & \\ x_{it} \geq 0 & \forall i \in A, t = 1, \ldots, T & (18) \end{array}$$

By posing $F = \max_{i \in A} \left(T - \sum_{t=1}^{T} f_{it} + 1 \right)$, this problem can be rewritten as:

$$\begin{array}{lll} \min F & & \\ x_{jt} \leq s_{i,t-1} + g_{ij}^1 & \forall (i,j) \in A_1, t = 1, \ldots, T & (1) \\ \ldots & \ldots & \\ \ldots & \ldots & \\ x_{it} \geq 0 & \forall i \in A, t = 1, \ldots, T & (18) \\ F \geq \left(T - \sum_{t=1}^{T} f_{it} + 1 \right) & \forall i \in A & (19) \end{array}$$

Proposition 1 *Minimizing the completion time of a set of activities under feeding precedence constraints and unlimited resources is polynomially solvable.*

Proof: The thesis comes out by observing that

$$\sum_{t=1}^{T} f_{it} = \max_{t=1,\ldots,T} (T - t) f_{it},$$

that substituted in constraints (19) gives:

$$F \geq \left(T - \max_{t=1,\ldots,T} (T - t) f_{it} + 1 \right) \forall i \in A.$$

Denoting

$$F_i = \max_{t=1,\dots,T} (T-t) f_{it}$$

we have formulation can be rewritten as

$$\begin{array}{lll}
\min F & & \\
x_{jt} \le s_{i,t-1} + g^1_{ij} & \forall (i,j) \in A_1, t = 1,\dots,T & (1) \\
\dots & \dots & \\
\dots & \dots & \\
x_{it} \ge 0 & \forall i \in A, t = 1,\dots,T & (18) \\
F \ge (T - F_i + 1) & \forall i \in A & (19') \\
F_i \ge (T-t) f_{it} & \forall i \in A, t = 1,\dots,T & (19'').
\end{array}$$

In the latter formulation we have at most two variables for each constraint, and therefore it is polynomially solvable (see, e.g., Hochbaum and Naor, 1994). □

2.2 The Scenario with Scarce Resources: a Lower Bound Calculation

When resources are limited, we have to introduce an additional constraints taking into account the resource availability. In detail, we assume that K renewable resources are available in amounts of b_k units, with $k = 1,\dots,K$. Each activity $i \in A$ has to be carried out by using q_{ik} units of resource $k = 1,\dots,K$. The formulation P of the problem with resource constraints is therefore:

$$\begin{array}{lll}
\min F & & \\
x_{jt} \le s_{i,t-1} + g^1_{ij} & \forall (i,j) \in A_1, t = 1,\dots,T & (1) \\
\dots & \dots & \\
\dots & \dots & \\
F \ge \left(T - \sum_{t=1}^T f_{it} + 1\right) & \forall i \in A & (19) \\
\sum_{i=1}^{|A|} q_{ik}(x_{it} - x_{i,t-1}) \le b_k & k = 1,\dots,K, t = 1,\dots,T & (20)
\end{array}$$

where constraints (20) impose that, in each time slot and for each resource type, the sum of the overall resource requirement does not exceed the total availability.

Proposition 2 *Minimizing the completion time of a set of activities under feeding precedence constraints and limited resources is NP-hard.*

Proof: We know, from the state of the art, that problem P', obtained by adding to P the classical (finish-to-start) precedence constraints, is difficult (see, e.g., Kiss, 2006). Assume to write P inserting the set of precedence constraints (PC); we have the following formulation of P':

$$
\begin{array}{lll}
\min F & & \\
x_{jt} \leq s_{i,t-1} + g^1_{ij} & \forall (i,j) \in A_1, t = 1, \ldots, T & (1) \\
\ldots & \ldots & \\
\ldots & \ldots & \\
\sum_{i=1}^{|A|} q_{ik}(x_{it} - x_{i,t-1}) \leq b_k & k = 1, \ldots, K, t = 1, \ldots, T & (20) \\
s_{j,t} \leq x_{i,t-1} & \forall (i,j) \in PC, t = 1, \ldots, T & (21)
\end{array}
$$

Note that precedence constraints (21) are a special case of feeding precedence constraints (2) with $q^1_{ij} = 1$, i.e., the set PC can be interpreted as a subset in A_2 with $q^1_{ij} = 1, \forall (i,j) \in PC$. Therefore, P is as difficult as P', i.e., it is NP-hard. □

Starting from this complexity result, in the following, we pose the goal of calculating a good lower bound to the optimal solution of P. We will tackle this objective by showing the Lagrangian relaxation of constraints (20) in problem P.

The Lagrangian model P_{LaR} is as follows:

$$
\begin{array}{lll}
\min \left\{ F \; - \; \sum_{k=1}^{K} \sum_{t=1}^{T} \right. & \left. \lambda_{kt} \left[b_k - \sum_{i=1}^{|A|} q_{ik}(x_{it} - x_{i,t-1}) \right] \right\} & \\
x_{jt} \leq s_{i,t-1} + g^1_{ij} & \forall (i,j) \in A_1, t = 1, \ldots, T & (1) \\
\ldots & \ldots & \\
\ldots & \ldots & \\
F \geq \left(T - \sum_{t=1}^{T} f_{it} + 1 \right) & \forall i \in A & (19)
\end{array}
$$

where for each value of the λ_{kt} multipliers, the optimal solution of the model P_{LaR} provides a lower bound to the optimal solution of problem P. The dual Lagrangian model is to find the best λ_{kt} values to maximize the following problem

$$
\begin{array}{lll}
\max_{\lambda_{kt}} \left\{ \min \left\{ F \; - \; \sum_{k=1}^{K} \right. \right. & \left. \left. \sum_{t=1}^{T} \lambda_{kt} \left[b_k - \sum_{i=1}^{|A|} q_{ik}(x_{it} - x_{i,t-1}) \right] \right\} \right\} & \\
x_{jt} \leq s_{i,t-1} + g^1_{ij} & \forall (i,j) \in A_1, t = 1, \ldots, T & (1) \\
\ldots & \ldots & \\
\ldots & \ldots & \\
F \geq \left(T - \sum_{t=1}^{T} f_{it} + 1 \right) & \forall i \in A & (19)
\end{array}
$$

By constraints (15) and (19) we have $x_{it} \geq f_{it}$ and $F \geq (T - \sum_1^T f_{it} + 1)$; therefore,

$$F \geq (T - \sum_1^T x_{it} + 1)$$

and, hence,

$$\sum_1^T x_{it} \geq (T - F + 1) \;\; \text{(a)}.$$

If we multiply both the left- and right-hand sides of (a) by $\lambda_{kt} q_{ik}$, and sum up with respect to i, k, t, we get the following relation:

$$\sum_{i=1}^{|A|} \sum_{k=1}^{K} \sum_{t=1}^{T} \lambda_{kt} q_{ik} x_{it} \geq (T - F + 1) \sum_{i=1}^{|A|} \sum_{k=1}^{K} \sum_{t=1}^{T} \lambda_{kt} q_{ik} \;\; \text{(b)}.$$

Rewriting the objective function accordingly, we have:

$$\max_{\lambda_{kt}} \left\{ -\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_{kt}b_k + \min\left[F + \sum_{i=1}^{|A|}\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_{kt}q_{ik}x_{it} - \sum_{i=1}^{|A|}\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_{kt}q_{ik}x_{i,t-1}) \right]\right\} \leq$$

$$\leq \max_{\lambda_{kt}} \left\{ -\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_{kt}b_k + \min\left[F + (T-F+1)\sum_{i=1}^{|A|}\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_{kt}q_{ik} - \sum_{i=1}^{|A|}\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_{kt}q_{ik}) \right]\right\},$$

where the second expression is obtained by minoring, inside the min term, the triple summation with the positive sign by exploiting (b), and majoring the triple summation with the negative sign, by posing $x_{i,t-1} = 1$.

Now we are in position to find an estimate $\tilde{\lambda}_{kt}$ of the optimal values λ^*_{kt} of the Lagrangian multipliers by solving the following problem:

$$\max_{\lambda_{kt}} \left\{ -\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_{kt}b_k + \min\left[F\left(1 - \sum_{i=1}^{|A|}\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_{kt}q_{ik}\right) + \sum_{i=1}^{|A|}\sum_{k=1}^{K}\sum_{t=1}^{T}T\lambda_{kt}q_{ik}\right]\right.$$
$$F \geq 0.$$
$$\lambda_{kt} \geq 0, \quad k = 1, \ldots, K, t = 1, \ldots, T$$

In fact, noting that the above problem admits a bounded solution if

$$\left(1 - \sum_{i=1}^{|A|}\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_{kt}q_{ik}\right) \geq 0.$$

the estimate $\tilde{\lambda}_{kt}$ of the Lagrangian multipliers is obtained by optimally solving the following linear programming:

$$\begin{aligned}
&\max_{\lambda_{kt}} \textstyle\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_{kt}\left[\sum_{i=1}^{|A|}Tq_{ik} - b_k\right] \\
&\textstyle\sum_{i=1}^{|A|}\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_{kt}q_{ik} \leq 1 \\
&\lambda_{kt} \geq 0, \quad k = 1, \ldots, K, t = 1, \ldots, T
\end{aligned}$$

The lower bound is therefore obtained by optimally solving the following problem:

$$\begin{array}{lll}
\min\left\{F - \sum_{k=1}^{K}\sum_{t=1}^{T}\right. & \left.\tilde{\lambda}_{kt}\left[b_k - \sum_{i=1}^{|A|}q_{ik}(x_{it} - x_{i,t-1})\right]\right\} & \\
x_{jt} \leq s_{i,t-1} + g^1_{ij} & \forall (i,j) \in A_1, t = 1, \ldots, T & (1) \\
\ldots & \ldots & \\
\ldots & \ldots & \\
F \geq \left(T - \sum_{t=1}^{T} f_{it} + 1\right) & \forall i \in A & (19)
\end{array}$$

This problem can be shown to be polynomially solvable by using the same arguments as those used in Proposition 1.

3 Computational Results

3.1 Implementation details

The experimentation on the lower-bound formulation P_{LaR} has been carried out by implementing the latter in the C and the AMPL language. The AMPL code have been solved by means of the CPLEX solver, version 8.0.0. The machine used for the experiments is a PC Core Duo with a 1.6 GHz Intel Centrino Processor and 1 GB RAM.

Experiments have been generated with the following features:

- the number of activities $|A|$ has been chosen equal to 10, 20, 30, 40, 50, 80 and 100;
- a density fd of feeding precedence constraints has been selected equal to 10%, 30% and 50%;
- the number K of renewable resources has been kept equal to 4;
- an amount b_k of resource availability per period for each resource $k = 1, \dots, K$ has been set equal to 4;
- a request q_{ik} of resource $k = 1, \dots, K$ for every activity $i \in A$ has been assigned uniformly at random from 1 to 2, from 1 to 3, and from 1 to 4;
- the values $g_{ij}^1, q_{ij}^1, g_{ij}^2, q_{ij}^2$ have been assigned uniformly at random in the range $(0.00, 1.00)$.

3.2 Analysis of the results

$fd = 10\%$, $q_{ik} \simeq unif(1,2)$										
$\|A\|$	IM	LiR	RR	FR	LaR	IM	LiR	RR	FR	LaR
	Average objective value					Objective function stand. dev				
10	4.400 (5/5)	2.235	2.900	3.400	3.742	1.06	0.60	1.15	0.89	0.67
20	7.400 (5/5)	3.754	5.000	5.100	6.792	0.70	0.82	0.39	0.96	1.41
30	8.200 (3/5)	4.146	5.600	5.860	7.289	5.11	1.02	0.67	0.43	1.56
40	18.400 (2/5)	6.287	8.800	8.780	12.894	4.80	0.68	0.73	0.61	1.42
50	24.600 (0/5)	8.541	12.000	11.900	16.782	5.09	1.10	1.53	0.84	2.98
80	34.800 (0/5)	11.486	17.400	17.200	26.328	4.20	2.42	0.77	0.89	0.87
100	45.000 (0/5)	15.422	20.800	21.000	32.556	5.18	2.55	1.00	0.48	2.06

Table 1: Comparison among objective values achieved by the integer model (IM), its linear relaxation (LiR), the integer model with resource relaxation (RR) and feeding constraints relaxation (FR), and the Lagrangian relaxation (LaR).

Results are shown in Tables 1-6, and are reported as averages over five instances; in particular, in Tables 1, 3 and 5 we reported the objective values of the integer model (IM),

$fd = 10\%$, $q_{ik} \simeq unif(1,2)$					
$\|A\|$	`IM`	`LiR`	`RR`	`FR`	`LaR`
	Average CPU time (sec.)				
10	0.644	0.810	0.146	0.000	0.028
20	62.338	7.058	0.177	0.000	0.031
30	1784.319	29.514	0.841	0.125	0.031
40	3127.139	166.417	1.464	0.196	1.203
50	3600.000	459.500	1.731	0.860	1.786
80	3600.000	1647.931	2.052	0.894	3.374
100	3600.000	3242.449	2.995	0.987	4.626

Table 2: Comparison among CPU times spent by the integer model (`IM`), its linear relaxation (`LiR`), the integer model with resource relaxation (`RR`) and feeding constraints relaxation (`FR`), and the Lagrangian relaxation (`LaR`).

its linear relaxation (`LiR`), the integer model with resource relaxation (`RR`) and feeding constraints relaxation (`FR`), and the Lagrangian relaxation (`LaR`). In Tables 2, 4 and 6, instead, we reported the CPU times.

Analyzing Table 1, where $fd = 10\%$ and q_{ik} follows a uniform probability distribution between 1 and 2, we notice that (see values into parentheses below column `IM`) only instances with 10 and 20 activities have been all solved to optimality by CPLEX. When $|A| = 30$ three out of five instances have been optimally solved within 3600 seconds of CPU time, and only two out of five in the case $|A| = 40$; instances with 50, 80 and 100 activities have not been solved at all within the time limit. This means that in the column `IM` values associated with 30, 40, 50, 80 and 100 activities are upper bounds on the minimum makespan.

Examining the values reported in column `LiR` it appears how for $|A| = 10, 20$, i.e., when all the instances are solved at the optimum, the lower bound obtained by the linear relaxation of `IM` is about 50% far from the optimum, highlighting how linear relaxation is a non-effective lower bounding strategy for this kind of problem. As soon as $|A|$ increases, as it can be expected, `LiR` tends to behave poorly, being the gap $\frac{\texttt{IM}-\texttt{LiR}}{\texttt{IM}} \simeq 66\%$ for $|A| \geq 40$.

Taking into account `RR`, i.e., the lower bound obtained by relaxing the resource constraints, the gap to `IM` values reduces. In particular, `RR` is able to improve on `LiR` values from a minimum of 30% (see $|A| = 10$) to a maximum of 52% (see $|A| = 80$), and the gap $\frac{\texttt{IM}-\texttt{RR}}{\texttt{IM}}$ ranges from 32% to 54%. Note that once the resource constraints have been relaxed the resulting problem is polynomially solvable as proved in Proposition 1. Its solution can be obtained, e.g., by applying the algorithm reported in Hochbaum and Naor (1994) or by using a commercial software as CPLEX, as we done. The result is that computational times are very limited, as reported in Table 2.

A similar behaviour can be observed taking into account the lower bound `FR` obtained by relaxing the feeding constraints and thus taking into account resource constraints only. FR can be computed in a closed form by the following simple formula

$$\max_{k=1,\dots,K} \sum_{i\in A} \frac{q_{ik}}{b_k}.$$

Due to the latter consideration, computing times are negligible, even with respect to RR.

Analyzing LaR behaviour, we note how it is always able to outperform the competing lower bounds used in this comparison. In particular, the percentage gap $\frac{\texttt{LaR}-\texttt{BestLb}}{\texttt{BestLb}}$, where BestLb is the best results taken among LiR, RR and FR, ranges from 10%, when $|A| = 10$, to 55% when $|A| = 100$. Moreover, the gap $\frac{\texttt{IM}-\texttt{LaR}}{\texttt{IM}}$ is 15% and 8% when $|A| = 10$ and $|A| = 20$, respectively, and ranges from 11% to 27% when $|A| \geq 30$.

Finally, we observe that computing times of LaR as negligible and comparable to those of RR.

$fd = 30\%$, $q_{ik} \simeq unif(1,3)$										
$\|A\|$	IM	LiR	RR	FR	LaR	IM	LiR	RR	FR	LaR
	Average objective value					Objective function stand. dev				
10	5.200 (5/5)	2.897	3.600	4.640	4.984	0.40	0.09	0.40	0.24	0.49
20	9.000 (5/5)	4.977	6.600	6.800	8.486	0.63	0.25	0.32	0.25	0.80
30	15.200 (2/5)	7.098	7.400	7.800	9.323	4.27	0.18	0.22	0.28	0.80
40	24.400 (1/5)	8.958	11.800	11.740	16.760	4.68	0.35	1.53	1.10	2.04
50	36.800 (0/5)	11.225	16.000	15.900	21.160	4.83	0.27	1.75	1.41	3.03
80	51.600 (0/5)	16.289	22.800	22.700	33.400	5.08	3.16	0.92	0.89	1.17
100	69.200 (0/5)	20.485	27.800	28.000	43.040	5.91	3.28	1.36	1.23	2.98

Table 3: Comparison among objective values achieved by the integer model (IM), its linear relaxation (LiR), the integer model with resource relaxation (RR) and feeding constraints relaxation (FR), and the Lagrangian relaxation (LaR).

$fd = 30\%$, $q_{ik} \simeq unif(1,3)$					
$\|A\|$	IM	LiR	RR	FR	LaR
	Average CPU time (sec.)				
10	0.650	0.872	0.150	0.000	0.031
20	67.606	7.656	0.182	0.000	0.031
30	1826.545	30.950	0.869	0.135	0.034
40	3214.540	177.806	1.532	0.202	1.316
50	3600.000	466.022	1.891	0.872	1.891
80	3600.000	1812.452	2.086	0.9442	3.438
100	3600.000	3591.460	3.124	1.038	4.637

Table 4: Comparison among CPU times spent by the integer model (IM), its linear relaxation (LiR), the integer model with resource relaxation (RR) and feeding constraints relaxation (FR), and the Lagrangian relaxation (LaR).

Analyzing Table 3, we notice that, as soon as the degree of freedom of activities decreases (here $fd = 30\%$ and q_{ik} follows a uniform probability distribution between 1 and 3), the number of instances solved at the optimum by CPLEX for $|A| = 30$ and $|A| = 40$ reduces, i.e., two out of five instances have been optimally solved within 3600 seconds of CPU time, and only one out of five in the case $|A| = 40$; instances with 50, 80 and 100 activities clearly remain optimally unsolved within the time limit, and one can expect that the upper bound quality is lowered by the increased difficulty of the instances.

Values reported in column LiR follow a trend similar to that observed for Table 1. These values are about 50% of the optimum when $|A| = 10$ and $|A| = 20$, and range from 53% to 72% when $|A| \geq 30$.

RR and FR have a similar behavior, as observed in Table 1. In particular, the gap $\frac{\texttt{IM}-\texttt{RR}}{\texttt{IM}}$ ranges from 27% to 60%, and the gap $\frac{\texttt{IM}-\texttt{FR}}{\texttt{IM}}$ ranges from 10% to 60%. Computing times of RR and FR remain very limited.

The Lagrangian relaxation LaR, in this more constrained scenario, is always able to outperform the competing lower bounds. In particular, the percentage gap $\frac{\texttt{LaR}-\texttt{BestLb}}{\texttt{BestLb}}$ is 7%, when $|A| = 10$, and is 54% when $|A| = 100$. Moreover, the gap $\frac{\texttt{IM}-\texttt{LaR}}{\texttt{IM}}$ is 4% and 6% when $|A| = 10$ and 20, respectively, and is about 36% when $|A| \geq 30$. Computing times of LaR remain negligible.

$fd = 50\%$, $q_{ik} \simeq unif(1, 4)$										
$\|A\|$	IM	LiR	RR	FR	LaR	IM	LiR	RR	FR	LaR
	Average objective value					Objective function stand. dev				
10	7.400 (5/5)	3.442	4.800	5.800	6.276	0.44	0.16	0.73	0.32	0.74
20	10.800 (5/5)	5.235	8.400	8.500	10.049	0.74	0.46	0.44	0.32	1.05
30	19.200 (1/5)	6.612	9.200	9.760	12.298	5.74	0.33	0.28	0.48	1.21
40	30.800 (0/5)	10.821	15.200	14.680	19.562	5.71	0.36	1.95	1.73	2.32
50	41.800 (0/5)	14.012	20.000	19.890	26.942	6.71	0.52	3.48	2.18	3.62
80	60.800 (0/5)	19.189	29.000	28.380	33.692	6.58	4.89	0.92	1.60	2.01
100	89.800 (0/5)	25.255	34.200	35.000	54.556	6.68	4.47	2.60	1.54	3.01

Table 5: Comparison among objective values achieved by the integer model (IM), its linear relaxation (LiR), the integer model with resource relaxation (RR) and feeding constraints relaxation (FR), and the Lagrangian relaxation (LaR).

Finally, we examine Table 5, where $fd = 50\%$ and q_{ik} follows a uniform probability distribution between 1 and 4. The trend is that now we are not able to solve any instances with 40 activities and only one out of five instances is solvable with 30 activities within the time limit.

The columns LiR, RR and FR show similar behavior patterns with respect to the previous two scenarios, being the linear relaxation the poorest among the three approaches, and RR and FR comparable, with a gap $\frac{\texttt{IM}-\texttt{RR}}{\texttt{IM}}$ ranging from 35% to 62%, and a gap $\frac{\texttt{IM}-\texttt{FR}}{\texttt{IM}}$ ranging from 22% to 61%. Running times of RR and FR are modest as one can expect.

$fd = 50\%$, $q_{ik} \simeq unif(1,4)$					
$\lvert A \rvert$	IM	LiR	RR	FR	LaR
	Average CPU time (sec.)				
10	0.675	0.958	0.150	0.000	0.032
20	69.331	8.120	0.197	0.000	0.031
30	1924.402	31.106	0.892	0.145	0.034
40	3600.000	181.435	1.573	0.208	1.389
50	3600.000	498.157	1.937	0.899	2.013
80	3600.000	1871.226	2.291	1.018	3.761
100	3600.000	3598.157	3.325	1.060	4.911

Table 6: Comparison among CPU times spent by the integer model (IM), its linear relaxation (LiR), the integer model with resource relaxation (RR) and feeding constraints relaxation (FR), and the Lagrangian relaxation (LaR).

LaR offers again the best lower bound values with a percentage gap $\frac{\texttt{LaR}-\texttt{BestLb}}{\texttt{BestLb}}$ ranging from 8% to 56%. The percentage gap $\frac{\texttt{IM}-\texttt{LaR}}{\texttt{IM}}$ is equal to 16%, when $|A| = 10$, and 7% when $|A| = 20$. Moreover, the gap $\frac{\texttt{IM}-\texttt{LaR}}{\texttt{IM}}$ ranges from 36% to 39% when $|A| \geq 30$. Running times of LaR range from 0.74 seconds to 3.62 seconds.

For the sake of completeness, in Figure 5 we report the gap $\frac{\texttt{IM}-\texttt{LaR}}{\texttt{IM}}$ over increasing density of feeding constraints, for the case in which q_{ik} follows a uniform probability distribution between 1 and 3. This has been conducted for four values of $|A|$, i.e., 10, 15, 20 and 25. These values have been selected as those for which, given a larger CPU time limit equal to 2 hours, CPLEX was able to optimally solve at least three out of five instances, and values are reported as averages computed only on these optimal values. Reading over the chart, it appears that the trend of the gap, even though it remains quite limited, is increasing with the feeding constraints density. This is what one could expect, since, when the effect of the resources is overwhelmed by the effect of the feeding constraints, the resources penalization term of LaR tends to be less effective.

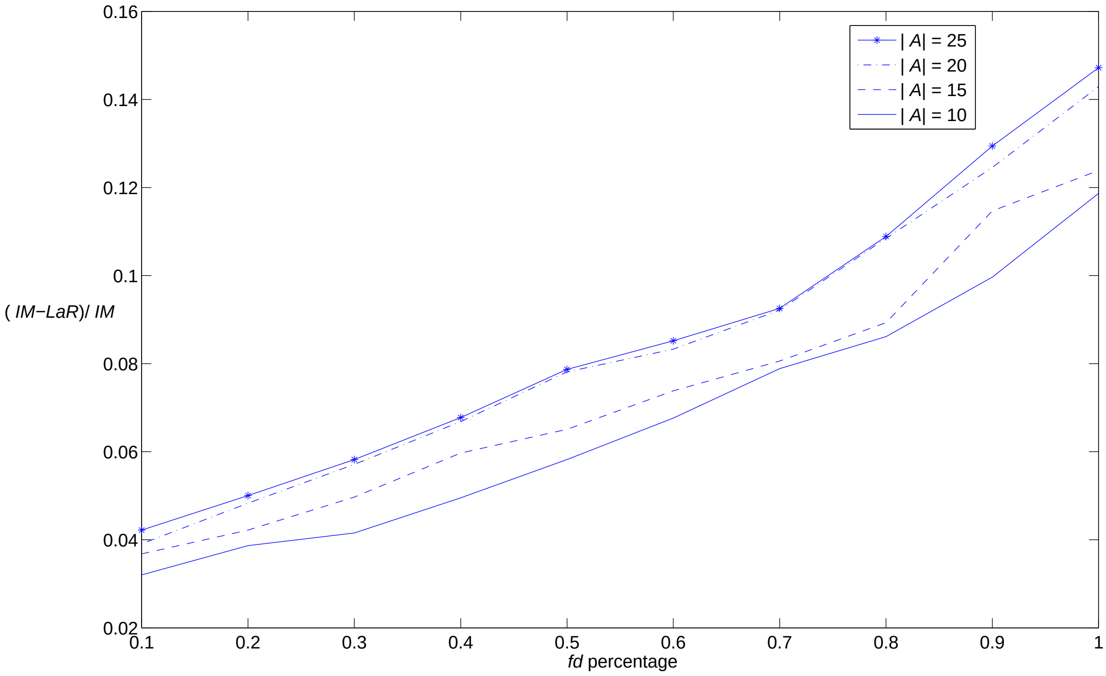

Figure 5: Percentage gap between the optimum values of `IM` and `LaR` over increasing density of feeding constraints ($q_{ik} \simeq unif(1,3)$).

4 Conclusions

We studied the problem of scheduling a set of activities under feeding precedence constraints, scarce resources and minimum makespan objective function. The problem is NP-hard and a lower bound based on a Lagrangian relaxation of the resource constraints has been proposed. Calculating a high quality lower bound value in a hard combinatorial problem with minimum objective functions is an important issue, and its importance is more appreciable when the problem under study models a real application. This is the case of project management where the importance of a good lower bound on the minimum makespan has an important practical interest. Indeed, even if a lower bound procedure does not return a feasible schedule, it allows the decision maker to verify if a given deadline suggested by the project commitment may be realistic or should be discarded a-priori because it results lower than the lower bound calculated. Therefore, it comes out that a poor lower bounding procedure may lead to the acceptance of a deadline that, even if greater than such lower bound, is lower than the shortest possible schedule length, i.e., the minimum makespan. We experimented with randomly generated instances of sizes of up to 100 activities, comparing the quality of the Lagrangian relaxation lower bound to that produced by linear relaxation, resource relaxation and feeding constraints relaxation. Tests have been carried out for different densities of feeding and resource constraints, and in all these scenarios we noticed a better performance of the proposed lower bound with respect to the competing ones. Future work will be devoted to develop an exact algorithm.

References

[1] Alfieri, A., T. Tolio, M. Urgo. 2007. Project Scheduling with Feeding Precedence Relations: an Application to Production Planning. In Proceedings of 9th Biennial ASME Conference on Engineering Systems Design and Analysis ESDA 2008 July 7-9, 2008, Haifa, Israel.

[2] Bartusch, M., R.H. Mohring, F.J. Radermacher. 1988. Scheduling Project Networks with Resource Constraints and Time Windows, Annals of Operations Research 16, 201-240.

[3] Bianco, L., M. Caramia. 2007. A New Formulation of the Resource-Unconstrained Project Scheduling Problem with Generalized Precedence Relations to Minimize the Completion Time. Technical Report DII - University of Rome "Tor Vergata", submitted.

[4] Demeulemeester, E.L., W.S. Herroelen. 2002. *Project Scheduling - A Research Handbook*. Kluwer Academic Publishers, Boston.

[5] De Reyck, B. 1998. Scheduling Projects with Generalized Precedence Relations - Exact and Heuristic Approaches. Ph.D. Thesis, Department of Applied Economics, Katholieke Universiteit Leuven, Leuven, Belgium.

[6] Dorndorf, U. 2002. *Project Scheduling with Time Windows*. Physica-Verlag, Heidelberg.

[7] Elmaghraby, S.E.E., J. Kamburowski. 1992. The Analysis of Activity Networks under Generalized Precedence Relations (GPRs). Management Science **38(9)** 1245–1263.

[8] Hochbaum, D., J. Naor. 1994. Simple and Fast Algorithms for Linear and Integer Programs with Two Variables per Inequality, SIAM Journal on Computing **23(6)** 1179–1192.

[9] Kis, T., G. Erdõs, A. Márkus, J. Váncza. 2004. A Project-Oriented Decision Support System for Production Planning in Make-to-Order Manufacturing, ERCIM news **58** 66–67.

[10] Kis, T. 2005. A branch-and-cut algorithm for scheduling of projects with variable-intensity activities, Mathematical Programming **103(3)** 515–539.

[11] Kis, T. 2006. Rcps with variable intensity activities and feeding precedence constraints. In Perspectives in Modern Project Scheduling, Springer US, pp. 105-129.

[12] Neumann, K., C. Schwindt, J. Zimmerman. 2002. *Project Scheduling with Time Windows and Scarce Resources*. Lecture Notes in Economics and Mathematical Systems 508, Springer Verlag, Berlin.