



HAL
open science

A project scheduling problem with periodically aggregated resource-constraints

Pierre-Antoine Morin, Christian Artigues, Alain Haït, Tamás Kis, Frits C.R. Spieksma

► **To cite this version:**

Pierre-Antoine Morin, Christian Artigues, Alain Haït, Tamás Kis, Frits C.R. Spieksma. A project scheduling problem with periodically aggregated resource-constraints. *Computers and Operations Research*, 2022, 141, pp.105688. 10.1016/j.cor.2021.105688 . hal-03525068

HAL Id: hal-03525068

<https://laas.hal.science/hal-03525068v1>

Submitted on 13 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A project scheduling problem with periodically aggregated resource-constraints

Pierre-Antoine Morin^{1,2} Christian Artigues¹ Alain Haït^{2,1} Tamás Kis³
Frits C.R. Spieksma⁴

¹ LAAS CNRS, University of Toulouse, CNRS, Toulouse, France.

² ISAE SUPAERO, University of Toulouse, Toulouse, France.

³ MTA SZTAKI, Hungarian Academy of Sciences, Budapest, Hungary.

⁴ Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven,
The Netherlands.

Abstract

We consider the so-called periodically aggregated resource-constrained project scheduling problem. This problem, introduced by Morin et al. 2017, is a variant of the well-known resource-constrained project scheduling problem that allows for a more flexible usage of the resource constraints. While the start and completion times of the activities can be arbitrary moments in time, the limitations on the resource usage are considered on average over aggregated periods of parameterized length. This paper presents new theoretical and experimental results for this problem. First, we settle the complexity status of the problem by proving NP-hardness of a number of special cases of the problem. Second, we propose a new mixed-integer programming formulation of the problem by disaggregating the precedence constraints over the periods. A theoretical comparison shows that the new formulation dominates the previously proposed one in terms of relaxation strength. Finally, we carry out computational experiments on instances from the literature to compare the merits of the different formulations.

keywords: Project Scheduling; Periodic Aggregation of resource-constraints; Computational complexity; Mixed-integer linear programming

1 Introduction

In the extensively studied standard resource-constrained project scheduling problem (RCPSP), at any time, the sum of the requirements of the activities that are currently processed must not exceed the resource capacity. This scheme permits to generalize a wide range of scheduling problems. However, in some practical applications, the time horizon is divided uniformly into consecutive intervals, and only the *average* activity requirements on each interval is considered. This aggregated form of resource constraints appears notably in employee scheduling where the load generated by the different activities and its compatibility with the number of present employees is evaluated on average in each shift (Paul and Knust 2015). However, the schedule of the activities can, and often should be determined on a more precise time scale for specific reasons such as the necessary anticipation for the usage

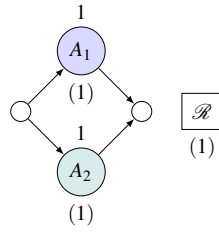
33 of scarce resources or the contractual relationships with suppliers and customers (Artigues et al. 2009). Another
34 example can be found in manufacturing or smart building applications, where the electricity consumption of jobs
35 is only computed in intervals fixed by the electricity provider while the schedule of the jobs can be more detailed
36 (Haït and Artigues 2011). In the literature, averaging the resource demand of activities inside fixed length periods
37 has been proposed for problems with variable-intensity activities: the rough cut capacity planning (RCCP) (Hans
38 2001) and the resource-constrained project scheduling problem with variable intensity activities (RCPSVP) (Kis
39 2005). An extension of the RCPSP with partially renewable resources, entitled RCPSP/II, has been introduced by
40 Böttcher et al. (1999), that allows to define intervals with specific rules for resource consumption. However, the
41 formulations proposed by Hans for the RCCP and by Kis for the RCPSVP do not involve variables representing
42 start times; moreover, there is no assumption (a priori) nor algorithm (a posteriori) that provides values for start
43 times. However, the average “energy” (duration \times demand) is an explicit variable. For a given solution (average
44 energy of each activity on each resource in each period), resource constraints induce bounds on start times for
45 compatible schedules (i.e. schedules whose energy profile matches the solution). In some cases, any such schedule
46 is precedence-infeasible. For the RCCP, a workaround has been proposed, but might fail. For the RCPSVP, in order
47 to avoid this phenomenon, for each predecessor/successor pair, the standard end-to-start precedence constraint is
48 replaced with the following constraint: if the predecessor completes in period ℓ , then the successor may start only
49 in period $\ell + 1$ or later, which leads to overconstrained precedence constraints, compared to the standard ones. In
50 both cases, no start times are involved. Apart from being undesirable for the above-mentioned specific reasons,
51 generally, this enforcement also has a strong negative impact on the scheduling objectives.

52 For the RCPSP/II, start times must coincide with sub-interval bounds and so can be seen as discrete variables:
53 because of this, not only optimal but also feasible solutions, possibly even all of them, may be excluded.

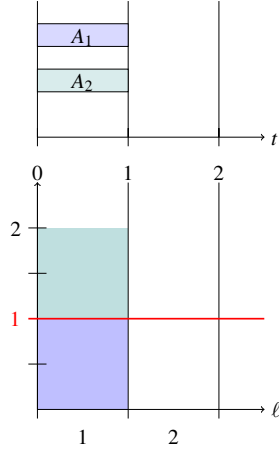
54 This paper focuses on the Periodically Aggregated Resource-Constrained Project Scheduling Problem
55 (PARCPSP), introduced in Morin et al. (2017), that permits to address this modification of resource constraints,
56 while considering the activity start times as continuous variables. In organizations, this model allows to consider
57 decisions at an intermediate level between planning at the tactical level for the resource limitation constraints and
58 scheduling at the operational level for time windows and precedence constraints.

59 Let us consider the following example (cf. figure 1), with a project composed of two activities and one resource
60 (cf. figure 1a). Each activity has one unit processing time and a unit resource consumption, while the resource has
61 one unit capacity. We suppose that the activities are not subject to a precedence relation. In the case of the RCPSP,
62 the resource is disjunctive: the standard cumulative resource constraint forbids that the activity execution windows
63 overlap, even partially. However, if these resource constraints are aggregated over time periods, e.g. of unit length,
64 such that the total average request should not exceed the resource capacity, then there exists feasible schedules such
65 that the two activities overlap, and even start and complete at the same moments in time. More precisely, among
66 such schedules, some remain infeasible (cf. figure 1b), while others become feasible (cf. figure 1c). This example
67 will be further commented in section 2.

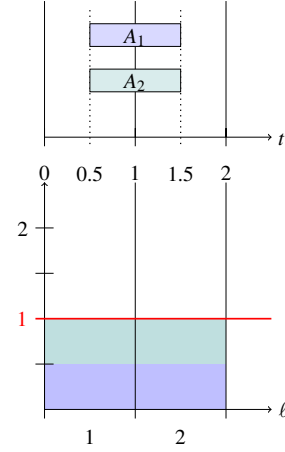
68 In Morin et al. (2017), a mixed-integer linear programming (MILP) formulation and heuristics are discussed,
69 while the problem itself is conjectured to be NP-complete. The aim of the current paper is, first, to establish



(a) Project instance



(b) Infeasible solution



(c) Feasible solution

Figure 1: Example #1

70 the NP-completeness of the PARCPSP, with various restrictions on the input, to highlight non-standard structural
 71 properties of this problem and, second, to propose alternative mixed-integer linear programming formulations with
 72 tighter relaxations.

73 The paper is structured as follows. In section 2, the PARCPSP is defined formally and compared to traditional
 74 resource-constrained project scheduling problems. In section 3, the PARCPSP is proved to be strongly NP-hard,
 75 by focusing on the computational complexity characterization of three particular cases. In section 4, the MILP
 76 formulation proposed in Morin et al. (2017) is recalled and a new formulation is proposed. The new formulation is
 77 shown to dominate the previous formulation in terms of LP relaxation. Computational experiments to compare the
 78 new formulation with the previous one are given in section 5. Finally, in section 6, some concluding remarks are
 79 drawn and possible extensions of the problem are discussed.

80 2 PARCPSP – Problem statement

81 In this section, we formally introduce the problem studied. It is a variant of the extensively studied Resource
 82 Constrained Project Scheduling Problem (RCPSP), based on a temporal aggregation of resource constraints over
 83 periods defining a uniform subdivision of the time horizon, hence the name Periodically Aggregated Resource
 84 Constrained Project Scheduling Problem (PARCPSP).

85 2.1 Input and notations

86 The input of the problem can be split into two independent parts.

87 • On the one hand, a project instance X is considered. An activity set and a resource set are given. Activities
 88 require a given amount of capacity on some or all resources throughout their execution. They cannot be
 89 interrupted: preemption is not allowed. Precedence relations possibly exist between activities.

90 The notations related to the project instance are listed hereafter.

\mathcal{A}	Finite set of n activities
\mathcal{R}	Finite set of m renewable resources
p_i	Processing time of activity $i \in \mathcal{A}$
b_k	Capacity of resource $k \in \mathcal{R}$
$r_{i,k}$	Request (demand) of activity $i \in \mathcal{A}$ on resource $k \in \mathcal{R}$
E	$\subseteq \mathcal{A} \times \mathcal{A}$; precedence relations (arc list)

91 Let \mathcal{X} the set of project instances.

92 • On the other hand, the time horizon is divided uniformly into L periods of parameterized length $\Delta \in \mathbb{R}_{>0}$. The
 93 convention chosen for period numbering is represented in figure 2.

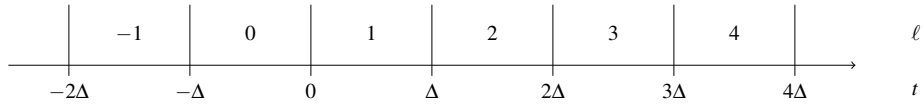


Figure 2: Uniform subdivision of the time horizon

94 A solution is a vector $S = (S_i)_{1 \leq i \leq n} \in \mathbb{R}^n$, where S_i is the start date of activity $i \in \mathcal{A}$. The start date and the
 95 completion date of the project are denoted by $S_0 = \min_{i \in \mathcal{A}}(S_i)$ and $S_{n+1} = \max_{i \in \mathcal{A}}(S_i + p_i)$, respectively.

96 2.2 A formulation of the PARCPSP

Let $S \in \mathbb{R}^n$ a solution of the PARCPSP. We consider two alternative objective functions linked to the temporal execution of the project.

$$C_{max} = S_{n+1} \quad (\text{project makespan})$$

$$dur(S) = S_{n+1} - S_0 \quad (\text{project duration})$$

97 Notice that, for the PARCPSP, unlike the RCPSP, these two objectives are not equivalent, because there is
 98 no guarantee that, for any of these objectives, at least one activity starts at $t = 0$ (hence $S_0 > 0$). This is further
 99 discussed in section 2.5. Moreover, if we set $S_0 \geq 0$, the problem defined with objective duration is indeed a
 100 relaxation of the one defined with objective makespan, since $S_{n+1} = C_{max}$.

101

102 Two families of constraints are taken into account.

103 1. Precedence constraints

104 For each arc $(i_1, i_2) \in E$, activity i_1 has to complete before activity i_2 starts.

105 2. Resource constraints (periodically aggregated)

106 For each resource $k \in \mathcal{R}$, in each period $\ell \in \mathbb{Z}$, the sum of the average requests of the activities cannot exceed
 107 the capacity of the resource.

Let $d_{i,\ell}(S) \in [0, \Delta]$ denote the execution duration of activity $i \in \mathcal{A}$ in period $\ell \in \mathbb{Z}$ depending on solution S , i.e., $d_{i,\ell}(S)$ is the length of the intersection of two intervals: the execution interval of activity i , and period ℓ (figure 3).

$$\begin{aligned} d_{i,\ell}(S) &= |[S_i, S_i + p_i] \cap [(\ell - 1)\Delta, \ell\Delta]| \\ &= \max\left(0, \min(S_i + p_i, \ell\Delta) - \max(S_i, (\ell - 1)\Delta)\right) \end{aligned}$$

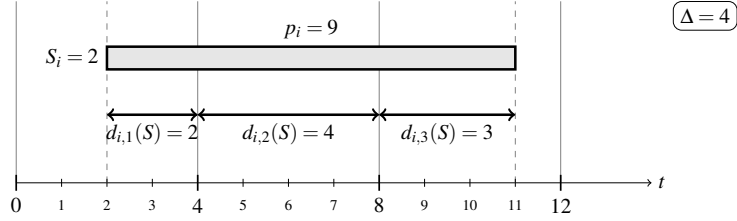


Figure 3: Evaluation of the execution duration in aggregated periods

Notice that, given a solution S , the expression of the average request of activity $i \in \mathcal{A}$ on resource $k \in \mathcal{R}$ over period $\ell \in \mathbb{Z}$ is $r_{i,k} \frac{d_{i,\ell}(S)}{\Delta}$. Therefore, the PARCPSP can be formulated as follows:

$$\text{Minimize } dur(S) \tag{1}$$

$$\text{subject to } S_{i_2} - S_{i_1} \geq p_{i_1} \quad \forall (i_1, i_2) \in E \tag{2}$$

$$\sum_{i \in \mathcal{A}} r_{i,k} \frac{d_{i,\ell}(S)}{\Delta} \leq b_k \quad \forall k \in \mathcal{R}, \forall \ell \in \mathbb{Z} \tag{3}$$

108 *Remark.* Activities may start at any time within a period. In other words, the PARCPSP permits to tackle start and
 109 completion events in a precise way, as well as precedence constraints, while the resource consumption is evaluated
 110 on average over (aggregated) periods.

111 *Remark.* Notice that the formulation is easily adjusted when capacities depend on the period, in which case b_k is
 112 replaced by $b_{k,\ell}$. Non uniform period length can similarly be obtained by replacing Δ by Δ_ℓ .

113 In the following, the notation PARCPSP[X, Δ] is used to identify the problem instance considered, composed of
 114 a project $X \in \mathcal{X}$ and a period length $\Delta \in \mathbb{R}_{>0}$. Similarly, the notation RCPSP[X] is used.

115 2.3 Conditions for the existence of feasible schedules

116 Let $X \in \mathcal{X}$ be a project instance and $\Delta \in \mathbb{R}_{>0}$ a (fixed) period length.

- 117 • The precedence constraints are satisfiable *iff* the precedence graph is acyclic.
- 118 • Let $i \in \mathcal{A}$. Let $k \in \mathcal{R}$. Let S denote a feasible solution of PARCPSP[X, Δ]. Let $\ell_i = 1 + \lfloor \frac{S_i}{\Delta} \rfloor$ denote the period
 119 in which activity i starts (i.e. such that $(\ell_i - 1)\Delta \leq S_i < \ell_i\Delta$).

120 – If $p_i \geq 2\Delta$

121 The execution window $[S_i, S_i + p_i]$ fully includes period $\ell_i + 1$. So, in this period: $r_{i,k} d_{i,\ell_i+1}(S) = r_{i,k} \Delta \leq$
 122 $b_k \Delta$. Hence: $r_{i,k} \leq b_k$.

123 – If $p_i < 2\Delta$

124 The least restrictive configuration is such that the middle of the execution window is a period bound
 125 (otherwise by shifting the activity in any direction, the maximum overlapping with the left or the right
 126 period increases, which increases the maximum resource requirement of the activity among all periods)
 127 i.e.: $S_i + \frac{p_i}{2} = \ell_i\Delta$. In this case, $d_{i,\ell_i}(S) = d_{i,\ell_i+1}(S) = \frac{p_i}{2}$ while $d_{i,\ell}(S) = 0$ in all other periods $\ell \in$
 128 $\mathbb{Z} \setminus \{\ell_i, \ell_i + 1\}$. In other words, the demand of activity i is split equally over two consecutive periods.
 129 So, the resource constraints in periods ℓ_i and $\ell_i + 1$ result in the same inequality: $r_{i,k} \frac{p_i}{2} \leq b_k\Delta$.

130 Hence: $r_{i,k} \leq b_k \frac{2\Delta}{p_i}$

Therefore, the (aggregated) resource constraints are satisfiable iff :

$$\forall i \in \mathcal{A} \quad \forall k \in \mathcal{R} \quad r_{i,k} \leq b_k \max \left\{ 1, \frac{2\Delta}{p_i} \right\}$$

131 **Remark** If the project instance X satisfies the following conditions, then, whatever the value of Δ (period length),
 132 there exist feasible schedules.

- 133 • The precedence graph is acyclic.
- 134 • $\forall i \in \mathcal{A} \quad \forall k \in \mathcal{R} \quad r_{i,k} \leq b_k$

135 2.4 Comparison with the RCPSP

A possible formulation for the RCPSP is:

$$\text{Minimize } dur(S) \tag{4}$$

$$\text{subject to } S_{i_2} - S_{i_1} \geq p_{i_1} \quad \forall (i_1, i_2) \in E \tag{5}$$

$$\sum_{i \in \mathcal{A}_t(S)} r_{i,k} \leq b_k \quad \forall k \in \mathcal{R}, \forall t \in \mathbb{R} \tag{6}$$

In this formulation, $\mathcal{A}_t(S)$ denote the set of activities in progress at $t \in \mathbb{R}$ depending on solution S .

$$\mathcal{A}_t(S) = \{i \in \mathcal{A} \mid t \in [S_i, S_i + p_i)\}$$

136 Notice that the only difference between the RCPSP and the PARCPSP lies in the definition of the resource
 137 constraints, which are evaluated either exactly at each instant $t \in \mathbb{R}$ or on average in each (aggregated) period $\ell \in \mathbb{Z}$
 138 (see figure 4), which makes the PARCPSP a relaxation of the RCPSP.

139 Another way of viewing this is to consider the impact of very small Δ . Indeed, if Δ becomes small, then, given
 140 the definition of $d_{i,\ell}(S)$, intervals ℓ in which the activity is processed will be completely occupied by the activity,
 141 and hence feature $d_{i,\ell}(S) = 1$. This means that, as Δ becomes smaller, formulation (1)-(3) converges to formulation
 142 (4)-(6).

143 2.5 Impact of aggregation on resource feasibility

144 Finally, let us consider two simple examples, respectively in figure 1 page 3 (example 1, already investigated in the
 145 introduction), and in figure 5 page 8 (example 2).

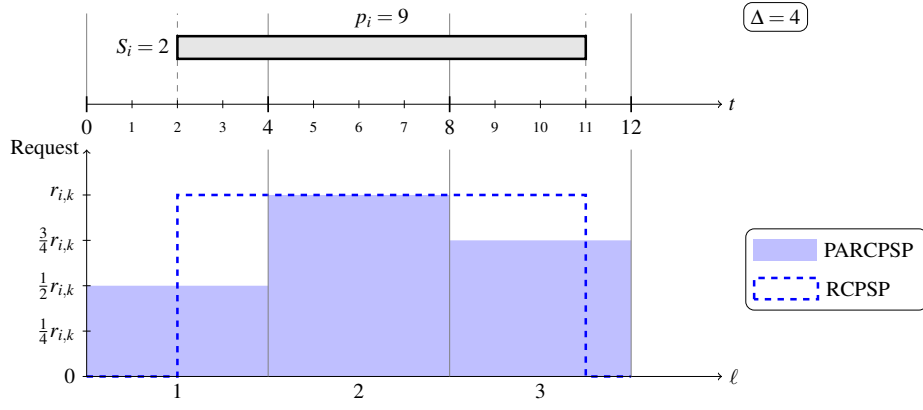


Figure 4: Evaluation of activity demands on resources

146 **Example 1** We recall that, in this example, the project instance $X_1 \in \mathcal{X}$ (see figure 1a) is composed of a single re-
 147 source of capacity 1 and two identical activities (same processing time equal to 1, same request on the resource equal
 148 to 1) with no precedence relations. We consider solutions such that the two activities are executed simultaneously,
 149 i.e. $S_1 = S_2$ (so, the project duration is equal to 1).

150 For the $\text{RCPSP}[X_1]$, such solutions are not feasible, since they violate resource constraints (evaluated at
 151 each instant). An optimal solution is obtained by executing one activity at a time, with no idle time; hence
 152 $\text{Opt}(\text{RCPSP}[X_1]) = 1 + 1 = 2$.

153 Let us consider a uniform subdivision of the temporal horizon, with periods of length $\Delta = 1$. What about the
 154 feasibility of such solutions for the $\text{PARCPSP}[X_1, 1]$?

155 • If $S_1 = S_2 = 0$, then both activities are completed at $t = 1$, i.e. both execution windows match exactly the first
 156 period ($\ell = 1$). In this period: $d_{1,1}(S) = d_{2,1}(S) = \Delta$. Therefore, the average request of each activity is equal
 157 to $\frac{\Delta}{\Delta} = 1$. So, as shown in figure 1b, the sum of the average requests in this period ($= 2$) exceeds the capacity
 158 of the resource ($= 1$). Hence this solution is not feasible.

159 • If $S_1 = S_2 = 0.5$, then both activities are completed at $t = 1.5$, i.e. both execution windows are split equally
 160 over two consecutive periods ($\ell \in \{1, 2\}$). In these periods: $d_{1,\ell}(S) = d_{2,\ell}(S) = \frac{\Delta}{2}$. Therefore, the average
 161 request of each activity is equal to $\frac{\Delta/2}{\Delta} = \frac{1}{2}$. So, as shown in figure 1c, the sum of the average requests in these
 162 periods ($= 1$) does not exceed the capacity of the resource ($= 1$). Hence this solution is feasible.

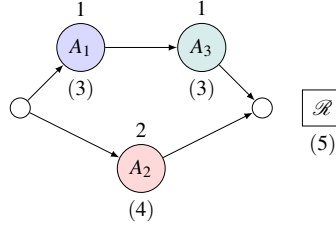
163 Indeed, this solution is optimal (since the two activities run in parallel, no other configuration can lead to a
 164 shorter project duration).

165 This first example enhances the following points.

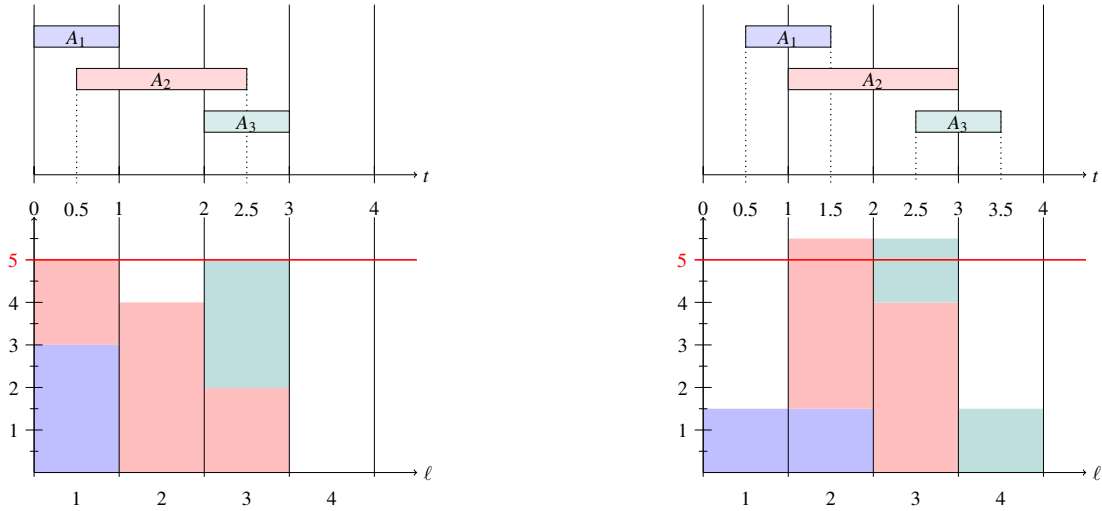
166 • Even when resources have a constant capacity over time, in the case of the PARCPSP, unlike the RCPSP,
 167 shifting a schedule can affect its feasibility.

168 • The gap between the optimum of the RCPSP and the PARCPSP can be large (here 50%) even with unit
 169 periods ($\Delta = 1$). In fact the example shows that the standard resource capacity lower bound equal to
 170 $\max_{k \in \mathcal{R}} \sum_{i \in \mathcal{A}} r_{i,k} p_i / b_k$ is not a valid lower bound for the PARCPSP.

171 For this particular instance, the project duration is reduced by dispatching the average requests equally over two
 172 consecutive periods. However, the rule “the more periods used, the shorter the project duration” does not apply to
 173 all instances, as shown in the next example.



(a) Project instance



(b) Feasible solution

(c) Infeasible solution

Figure 5: Example #2

174 **Example 2** The project instance $X_2 \in \mathcal{X}$ is composed of a single resource with capacity 5 and three activities with
 175 one precedence relation (see figure 5a for the numerical parameter values). We still consider unit periods ($\Delta = 1$).

176 • As shown in figure 5b, the solution $(0, 0.5, 2)$ is feasible. It is not optimal: one can shorten the project duration
 177 by shifting activity 1 to the right for an amount of $1/6$ and shifting activity 3 to the left for an amount of $1/6$,
 178 leading to a duration of 2.5.

179 Notice that 3 periods are intersected by at least one activity execution window.

180 • As shown in figure 5c, the solution $(0.5, 1, 2.5)$, obtained by shifting the previous solution by $+0.5$, is not
 181 feasible. It is possible to repair it, by shifting activities 1 and 3 by $-\frac{1}{6}$ and $+\frac{1}{6}$, respectively, thus enlarging
 182 the project duration by $\frac{1}{3}$ but now using 4 periods.

183 Therefore, we showed that the feasibility of a schedule depends not only on the relative positions of activity
 184 execution windows as in the RCPSP, but also on their absolute positions, which determines the average resource
 185 usage in aggregated periods. This problem has consequently fundamental differences with the related RCPSP.

186 3 Complexity

187 The complexity of the problem was left open in Morin et al. 2017. This section first establishes that the problem is
188 in NP, even if the time horizon is not part of the input. Then, the computational complexity of three particular cases
189 are considered, which yields the complexity result for the PARCPSP.

190 3.1 Inclusion in NP

191 To check whether a start time solution vector is feasible w.r.t. a fixed makespan requires checking the resource and
192 the precedence constraints. Since the number of time periods over a time horizon which covers both S_0 and S_{n+1}
193 may not be bounded by a polynomial in the size of the input, computing the average resource consumption in every
194 time period is not a viable approach to check the feasibility of a schedule in polytime. Testing resource constraints
195 only in periods where the resource usage increases is sufficient. Recall that activities may start at any time in a
196 period, and that preemption is not allowed. Therefore, each activity $i \in \mathcal{A}$ may increase the resource usage only in
197 two periods: the period when it starts ($\ell_i = 1 + \lfloor \frac{S_i}{\Delta} \rfloor$), and possibly the next one. So, for each activity, at most two
198 periods have to be checked. A single test in a given period on a given resource consists in verifying that the sum of
199 the mean demands of the activities is not greater than the capacity of the resource.

200 This yields an algorithm in $\mathcal{O}(1 + |E| + m \times 2n \times n)$, thus polynomial in the input size.

201 3.2 One resource, constant capacity

202 **Theorem 1.** *The PARCPSP with makespan objective and a single resource of fixed capacity $b \geq 2$ is weakly NP-*
203 *hard.*

204 *Proof.* We show that $\text{PARTITION} \leq_P \text{PARCPSP}$. In the PARTITION problem (Karp 1972), we have n items, and
205 each item i has a size $a_i \geq 1$. All the data is integral, and $\sum_i a_i$ is an even integer. Is there a partitioning of the items
206 into two subsets S_1 and S_2 , such that $S_1 \cap S_2 = \emptyset$, $S_1 \cup S_2 = \{1, \dots, n\}$, and $\sum_{i \in S_1} a_i = \sum_{i \in S_2} a_i$?

207 For any instance of the PARTITION problem, we define an instance of PARCPSP as follows. There is a single
208 resource of capacity 2. There are n activities, activity i corresponds to item i in the PARTITION problem instance,
209 and it has processing time $p_i := 2a_i$. The resource requirement of each activity is 1 from the single resource during
210 its execution. We let $\Delta = 1$. We claim that the PARTITION problem instance has a YES answer if and only if the
211 corresponding instance of PARCPSP admits a feasible schedule of length $(\sum_{i=1}^n p_i)/2$.

212 First suppose that the PARTITION problem instance has a YES answer. Then it must be the case that
213 $\sum_{i \in S_1} p_i = \sum_{i \in S_2} p_i = (\sum_{i=1}^n p_i)/2$. We define the following schedule: the activities corresponding to the items in
214 S_1 are scheduled in a single sequence from time 0 onwards. Notice that each activity starts and ends at integral
215 time points. This sequence occupies one unit of the resource from time 0 to time $(\sum_{i=1}^n p_i)/2$. Now schedule all the
216 activities corresponding to the items in S_2 in any sequence from time 0 onwards. Again, this sequence finishes at
217 time $(\sum_{i=1}^n p_i)/2$. Since $\Delta = 1$, the total capacity of the resource is 2 in each interval $[t - 1, t]$. Further on, in each in-
218 terval $[t - 1, t]$ with $t \leq (\sum_{i=1}^n p_i)/2$, the total resource usage is 2, because exactly two activities are processed in the
219 intervals, each requiring one unit from the resource. Therefore, the schedule is feasible, and all jobs are completed
220 by time $(\sum_{i=1}^n p_i)/2$.

221 Conversely, suppose there is a feasible schedule of length $(\sum_{i=1}^n p_i)/2$.

222 **Claim 1a.** *In any feasible schedule of length $(\sum_{i=1}^n p_i)/2$, exactly two units of resource are used in each interval.*

223 *Proof.* The total resource requirement of the activities is $\sum_{i=1}^n p_i$. Since the total capacity of the resource from time
 224 0 to time $(\sum_{i=1}^n p_i)/2$ is equal to $(\sum_{i=1}^n p_i)$, the claim follows. ■

225 **Claim 1b.** *Exactly two activities start at time 0.*

226 *Proof.* Suppose it is not the case. Observe that there can be at most two activities processed in the interval $[0, 1]$,
 227 because if there were 3 or more activities starting in the interval $[0, 1]$, then all these 3 or more activities should be
 228 processed throughout the interval $[1, 2]$, as each activity is of length 2 or more ($p_i = 2a_i$, and $a_i \geq 1$). But this is
 229 impossible, because the resource has capacity 2, and the activities would require 3 or more units of the resource.
 230 Now suppose that less than 2 activities start at time 0. Then the resource usage of the activities in interval $[0, 1]$ must
 231 be less than 2, which contradicts claim 1a. ■

232 So far we have shown that exactly two activities start at time 0 in the feasible schedule. Since the processing
 233 times are integral, these two activities finish at integral time points, at t_1 and t_2 , say. If $t_1 = t_2$, then we can repeat the
 234 same argument to show that there are exactly two activities starting right at time $t_1 = t_2$. If $t_1 \neq t_2$, then without loss
 235 of generality, $t_1 < t_2$. Then in the interval $[t_1, t_1 + 1]$, one unit of the resource is used by the activity which is still
 236 in progress. Since both t_1 and t_2 are divisible by 2 (as each p_i is divisible by 2), $t_2 \geq t_1 + 2$, and again, at most one
 237 activity may start in the interval $[t_1, t_1 + 1]$, otherwise in the interval $[t_1 + 1, t_1 + 2]$ the total resource usage would be
 238 more than 2. It follows that a new activity must be started at time t_1 , otherwise in the interval $[t_1, t_1 + 1]$, less than 2
 239 units of the resource would be used by the feasible schedule, which would contradict claim 1a. Proceeding in this
 240 way, we prove that all the activities start at integral time points, and at any time, at most two activities are processed.
 241 Hence, the schedule can be decomposed into two sequences of activities, S_1 and S_2 , each of the same total length
 242 $(\sum_{i=1}^n p_i)/2$. Then S_1 and S_2 give rise to a partitioning of the items such that $\sum_{i \in S_1} a_i = \sum_{i \in S_2} a_i$. Hence, the instance
 243 of the PARTITION problem has answer YES. □

244 The reduction does not hold for the duration $(S_{n+1} - S_0)$ objective. Consider the simple multiset of 5 elements
 245 $\{1, 1, 1, 1, 1\}$. Obviously, there is no partition of such set. If we now consider the PARCPSP obtained by the
 246 reduction, we obtain a set of 5 tasks of duration 2, each having a unit resource requirement on a resource of capacity
 247 2 and we also have $\Delta = 1$. Fig. 6 displays a feasible schedule of duration $S_{n+1} - S_0 = 5 = \sum_{i=1}^5 p_i/2$.

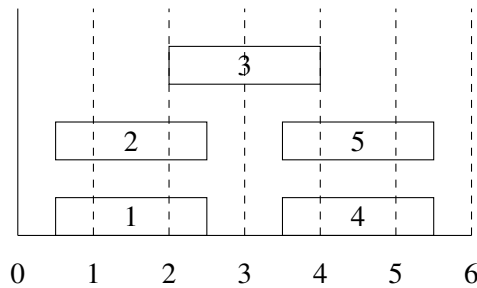


Figure 6: Feasible schedule, no partition

248 3.3 One resource, arbitrary capacity

249 The previous reduction (from the PARTITION problem) can be transformed slightly to derive the strong NP-
250 hardness of the PARCPSP with objective makespan from the 3-PARTITION problem (Garey and Johnson 1979),
251 when considering a single resource with arbitrary capacity.

252 **Theorem 2.** *The PARCPSP with makespan objective and a single resource is strongly NP-hard when capacity b is*
253 *part of the input.*

254 *Proof.* Let us show that 3-PARTITION \leq_P PARCPSP. Given $3n$ items of integral size a_i such that $\sum_i a_i = nD$, and
255 $\frac{D}{4} < a_i < \frac{D}{2}$ for all $i \in \{1, \dots, 3n\}$, the 3-PARTITION problem consists in determining whether a partitioning of the
256 items into n pairwise disjoint triples T_1, \dots, T_n of equal sum, i.e. $\sum_{i \in T_j} a_i = D$ for all $j \in \{1, \dots, n\}$, exist.

257 The reduction from the 3-PARTITION problem to PARCPSP is almost the same as the previous reduction from
258 PARTITION. Each item i is converted into an activity with processing time $p_i := 2a_i$, and a resource requirement
259 of 1; we only change the capacity of the single resource, setting it to n (previously set to 2).

260 Let us show that the 3-PARTITION problem instance has a YES answer if and only if the corresponding instance
261 of PARCPSP admits a feasible schedule of length $2D = (\sum_{i=1}^n p_i)/n$.

262 First suppose that the 3-PARTITION problem instance has a YES answer. A similar reasoning as the one
263 presented in the previous proof entails that scheduling the activities corresponding to a triple T_j in any order from
264 time 0 on in a single sequence yields a feasible schedule such that all jobs complete by time $2D$.

265 Conversely, suppose there is a feasible schedule of length $2D$. Claims 1a and 1b can be adapted seamlessly as
266 follows.

267 **Claim 2a** (generalization of claim 1a). *In each interval $[t-1, t]$ for $t \in \{1, \dots, 2D\}$, exactly n units of the resource*
268 *is used.*

269 **Claim 2b** (generalization of claim 1b). *Exactly n activities start at time 0.*

270 Moreover, there is no interval $[t-1, t]$, with $t \in \{1, \dots, 2D\}$, containing a moment during which less than n
271 activities are active (an activity being active at moment t if $S_j \leq t < S_j + p_j$). This can be seen using a contradiction
272 argument; suppose there is an interval containing a moment with less than n activities being active. Since, in this
273 interval, n units of resource must be used (after claim 2a), there must also be a moment in this interval in which more
274 than n activities are active. But that implies that a neighboring interval must feature more than n activities active
275 during that whole interval (since $p_j \geq 2$ and $\Delta = 1$), thereby exceeding the available capacity, which contradicts
276 claim 2a.

277 Therefore, at each instant in $[0, 2D]$, exactly n activities are active. Hence, the instance of the 3-PARTITION
278 problem has answer YES. □

279 3.4 Multiple resources, constant capacities

280 The third reduction, inspired from Blazewicz et al. (1983), establishes the strong NP-hardness of the PARCPSP
281 with objective duration or makespan for instances with an unlimited number of resources with constant capacities.

282 The proof presented hereafter considers the objective duration; notice that the proof for the objective makespan is
 283 very similar, because claim 3a holds regardless of the actual objective.

284 **Theorem 3.** *The PARCPSP with duration or makespan objective and unlimited number of resources with constant*
 285 *capacities is strongly NP-hard.*

286 *Proof.* We establish that Chromatic Number \leq_P PARCPSP. Given a non-oriented graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the Chromatic
 287 Number problem (Karp 1972) consists in coloring the vertices of \mathcal{G} using a minimum number of colors $(c_j)_{j \in \mathcal{V}}$ so
 288 that no two adjacent vertices are assigned the same color. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ a non-oriented graph. Let $\Delta \in \mathbb{R}_{>0}$ (e.g.
 289 $\Delta = 1$). Let $X(\mathcal{G}) \in \mathcal{X}$ the project instance defined by:

- 290 • $\mathcal{A} = \mathcal{V}$ (activity = vertex)
- 291 • $\mathcal{R} = \mathcal{E}$ (resource = edge)
- 292 • $\forall i \in \mathcal{A} \quad p_i = 2\Delta$
- 293 • $\forall k \in \mathcal{R} \quad b_k = 1$
- 294 • $\forall i \in \mathcal{A} \quad \forall k \in \mathcal{R} \quad r_{i,k} = 1$ if vertex i is one of the two extremities of edge k , 0 otherwise
- 295 • $E = \emptyset$ (no precedence relations)

296 Clearly, this is a polynomial time reduction; so, the theorem holds if the following assertions are equivalent.

297 1. \mathcal{G} admits a feasible coloring c such that:

$$298 \quad \max(c_j)_{1 \leq j \leq n} \leq \gamma$$

299 2. There exists a feasible schedule S such that:

$$300 \quad dur(S) \leq 2\gamma\Delta$$

Suppose \mathcal{G} admits a feasible coloring c such that $\max(c_j)_{1 \leq j \leq n} \leq \gamma$. Let S be the schedule defined by:

$$\forall i \in \mathcal{A} \quad S_i = 2(c_i - 1)\Delta$$

Given an edge (resource), its extremities (the two activities that require it) are colored differently (are not
 executed simultaneously, since processing times are all equal to 2Δ). So, S is feasible for the PARCPSP (indeed, it
 is even feasible for the RCPSP). Moreover:

$$dur(S) = S_{n+1} - S_0 \leq 2\gamma\Delta - 0 = 2\gamma\Delta$$

301 Hence, the direct implication holds.

302 Conversely, suppose there exists a feasible schedule S such that $dur(S) \leq 2\gamma\Delta$. Without loss of generality, the
 303 project execution starts in period $\ell = 1$, i.e., $0 \leq S_0 < \Delta$.

Claim 3a. *The execution windows of the two activities that share a common resource are disjoint.*

$$\forall (i_1, i_2) \in \mathcal{R} \quad (S_{i_1} + p_{i_1} \leq S_{i_2}) \vee (S_{i_2} + p_{i_2} \leq S_{i_1})$$

304 *Proof.* Let $k = (i_1, i_2) \in \mathcal{R}$. Suppose that $S_{i_1} \leq S_{i_2}$. For $i \in \mathcal{A}$, let $\ell_i = 1 + \lfloor \frac{S_i}{\Delta} \rfloor$ denote the period in which activity i
 305 starts. Notice that $\ell_{i_1} \leq \ell_{i_2}$.

For any activity $i \in \mathcal{A}$, including i_1 and i_2 , since $p_i = 2\Delta$, one can determine bounds on $d_{i,\ell}(S)$ (see also figure 7):

$$\forall \ell \in \mathbb{Z} \quad d_{i,\ell}(S) \begin{cases} \in (0, \Delta] & \text{if } \ell = \ell_i \\ = \Delta & \text{if } \ell = \ell_i + 1 \\ \in [0, \Delta) & \text{if } \ell = \ell_i + 2 \\ = 0 & \text{otherwise} \end{cases}$$

306 Indeed: $d_{i,\ell_i+2}(S) = \Delta - d_{i,\ell_i}(S)$

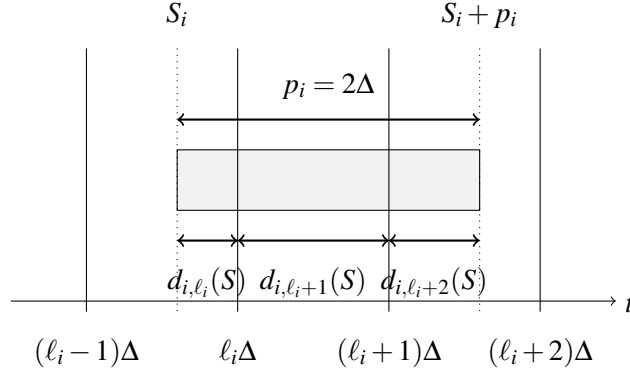


Figure 7: Execution interval (PARCPSP complexity proof)

Moreover, S is feasible; the resource constraints state that, in any period $\ell \in \mathbb{Z}$:

$$d_{i_1,\ell}(S) + d_{i_2,\ell}(S) \leq \Delta$$

307 • Suppose that: $\ell_{i_2} = \ell_{i_1}$

Then, in period $\ell = \ell_{i_1} + 1 = \ell_{i_2} + 1$:

$$d_{i_1,\ell}(S) + d_{i_2,\ell}(S) = 2\Delta > \Delta$$

308 Therefore, this configuration cannot occur.

309 • Suppose that: $\ell_{i_2} = \ell_{i_1} + 1$

Then, in period $\ell = \ell_{i_1} + 1 = \ell_{i_2}$:

$$d_{i_1,\ell}(S) + d_{i_2,\ell}(S) = \Delta + d_{i_2,\ell_{i_2}}(S) > \Delta$$

310 Therefore, this configuration cannot occur.

311 • Suppose that: $\ell_{i_2} = \ell_{i_1} + 2$

Then, in period $\ell = \ell_{i_1} + 2 = \ell_{i_2}$:

$$d_{i_1,\ell}(S) \leq \Delta - d_{i_2,\ell}(S)$$

$$\Leftrightarrow (\ell - 1)\Delta + d_{i_1,\ell}(S) \leq \ell\Delta - d_{i_2,\ell}(S)$$

$$\Leftrightarrow (\ell_{i_1} + 1)\Delta + d_{i_1,\ell_{i_1}+2}(S) \leq \ell_{i_2}\Delta - d_{i_2,\ell_{i_2}}(S)$$

$$\Leftrightarrow S_{i_1} + p_{i_1} \leq S_{i_2}$$

312 • Suppose that: $\ell_{i_2} \geq \ell_{i_1} + 3$

Then:

$$S_{i_1} + p_{i_1} < (\ell_{i_1} + 2)\Delta \leq (\ell_{i_2} - 1)\Delta \leq S_{i_2}$$

313 It follows that $S_{i_1} \leq S_{i_2} \Rightarrow S_{i_1} + p_{i_1} \leq S_{i_2}$. Hence, the claim holds. ■

Let c the coloring defined by:

$$\forall j \in \mathcal{V} \quad c_j = 1 + \left\lfloor \frac{S_j}{2\Delta} \right\rfloor$$

314 Let $(j_1, j_2) \in \mathcal{E}$. Recall that processing times are all equal to 2Δ ; so, after claim 3a, $|S_{j_2} - S_{j_1}| \geq 2\Delta$. By
 315 construction, $|c_{j_2} - c_{j_1}| \geq 1$ i.e. $c_{j_1} \neq c_{j_2}$. Therefore, c is feasible.

Since $dur(S) \leq 2\gamma\Delta$:

$$\forall i \in \mathcal{A} \quad S_0 \leq S_i \leq S_{n+1} - p_i \leq (S_0 + 2\gamma\Delta) - 2\Delta$$

316 Consequently, $1 \leq c_j \leq \gamma$ for all $j \in \mathcal{V}$, and $\max(c_j)_{1 \leq j \leq n} \leq \gamma$.

317 Hence, the reciprocal implication also holds. □

318 3.5 General case

319 The general result comes from the reductions provided for the three particular cases. In table 1, an arrow points to a
 320 more general/less restricted context for makespan minimization. Hence, the destination problem is at least as diffi-
 321 cult as the origin problem. The grey boxes correspond to complexity results holding also for duration minimization.

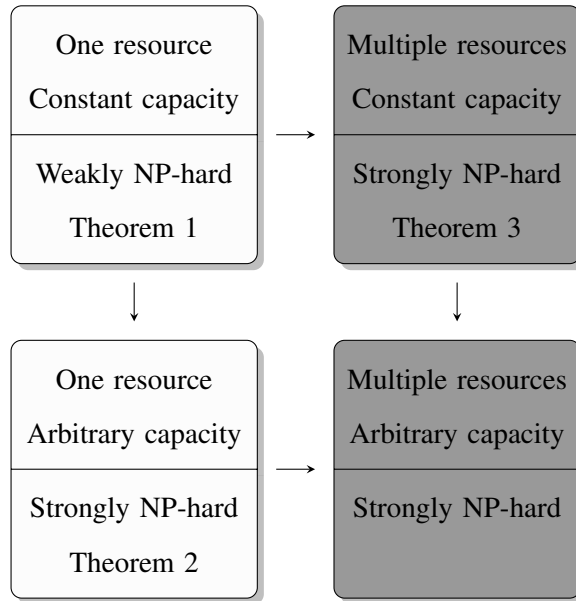


Table 1: Summary of the results on the NP-hardness of the PARCPSP for makespan minimization

322 It follows that the PARCPSP is strongly NP-hard in the general case. In the remaining of the paper, solution
 323 approaches are investigated.

324 4 A new mixed-integer linear programming formulation

325 In this Section, we consider mixed-integer linear programming formulations for the problem. Continuous variables
 326 are used to represent activity starting times while period-indexed variables allow to model the aggregated resource
 327 constraints. We consider two formulations and their strengthened variants. The first one was proposed by Morin
 328 et al. (2017) and the second one is a new formulation based on the decomposition of a period relatively to the
 329 execution of an activity. Both formulations can be strengthened by using bounds on the number of periods possibly
 330 intersected by an activity. In addition, the new formulation allows to use disaggregated precedence constraints. We
 331 show that the disaggregated second formulation is stronger than the first one in terms of LP relaxation.

332 4.1 First formulation

333 4.1.1 Variables

334 The decision variables used in the model proposed by Morin et al. (2017) are summarized in table 2. A continuous
 335 start time variable S_i gives the start time of each activity $i \in \mathcal{A}$ while a continuous variable $d_{i,\ell}$ gives the length of
 336 the intersection of the time window of activity $i \in \mathcal{A}$ with period $\ell \in \mathcal{L}$. Two period-indexed binary step variables
 337 $z_{s_{i,\ell}}$ and $z_{f_{i,\ell}}$ are used to mark the first and last periods of an activity. An illustration of the link between these
 338 variables is given in figure 8.

$S_i \geq 0$	Start time of Activity $i \in \mathcal{A}$ S_0 (resp S_{n+1}) represents the start (resp the end) of the project.
$d_{i,\ell} \in [0, \Delta]$	intersection length of intervals $[S_i, S_i + p_i]$ and $[(\ell - 1)\Delta, \ell\Delta]$
$z_{s_{i,\ell}} \in \{0, 1\}$	Binary <i>step</i> variables: $z_{s_{i,\ell-1}} \leq z_{s_{i,\ell}}$ $z_{s_{i,\ell}} = 1$ if S_i is in period ℓ , i.e. $S_i \in [(\ell - 1)\Delta, \ell\Delta]$
$z_{f_{i,\ell}} \in \{0, 1\}$	Binary <i>step</i> variables: $z_{f_{i,\ell-1}} \leq z_{f_{i,\ell}}$ $z_{f_{i,\ell}} = 1$ if $S_i + p_i$ is in period ℓ , i.e. $S_i + p_i \in [(\ell - 1)\Delta, \ell\Delta]$

Table 2: Variables of the first period-indexed formulation

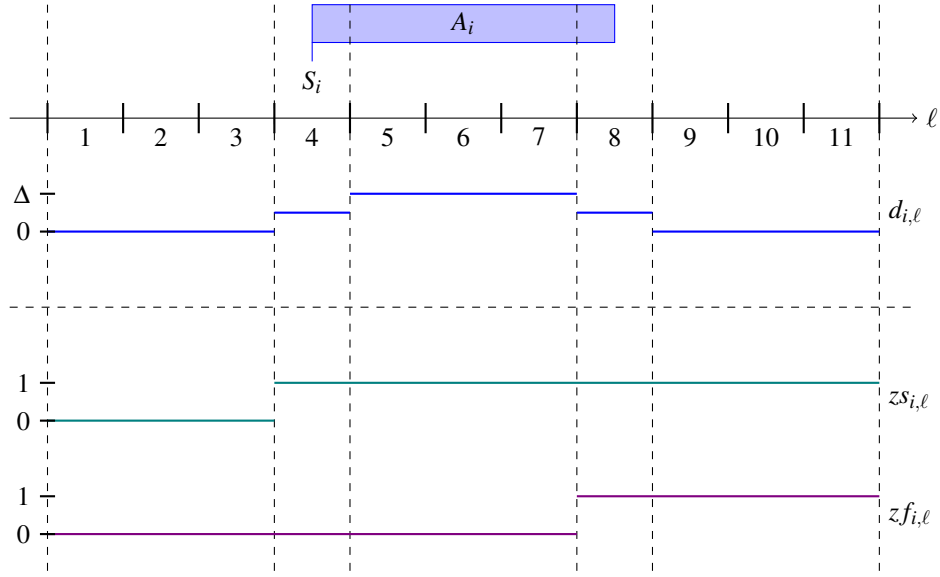


Figure 8: Representation of an execution time window with the variables of the first period-indexed formulation

339 4.1.2 Initial formulation

We recall below the main constraints of the formulation proposed by Morin et al. (2017), the domains of the decision variables being those of table 2.

$$(F1) \quad \text{Minimize} \quad S_{n+1} - S_0 \quad (7)$$

$$S_{i_2} - S_{i_1} \geq p_{i_1} \quad \forall (i_1, i_2) \in E \quad (8)$$

$$\sum_{i \in \mathcal{A}} r_{i,k} d_{i,\ell} \leq b_k \Delta \quad \forall k \in \mathcal{R}, \forall \ell \in \mathcal{L} \quad (9)$$

$$\ell \Delta (1 - z_{S_i,\ell}) \leq S_i \leq L \Delta - (L - \ell) \Delta z_{S_i,\ell} \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (10)$$

$$\ell \Delta (1 - z_{f_i,\ell}) \leq S_i + p_i \leq L \Delta - (L - \ell) \Delta z_{f_i,\ell} \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (11)$$

$$\Delta (z_{S_i,\ell-1} - z_{f_i,\ell}) \leq d_{i,\ell} \leq \Delta (z_{S_i,\ell} - z_{f_i,\ell-1}) \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (12)$$

$$d_{i,\ell} \geq \ell \Delta - S_i - \Delta z_{f_i,\ell} - \ell \Delta z_{S_i,\ell-1} \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (13)$$

$$d_{i,\ell} \geq S_i + p_i - (\ell - 1) \Delta - \Delta (1 - z_{S_i,\ell-1}) - (L - \ell + 1) \Delta (1 - z_{f_i,\ell}) \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (14)$$

$$\sum_{\ell \in \mathcal{L}} d_{i,\ell} = p_i \quad \forall i \in \mathcal{A} \quad (15)$$

340 Objective (7) minimizes the project duration, under precedence constraints (8) and aggregated resource con-
 341 straints (9). Constraints (10) link start time variables S_i and variables $z_{S_i,\ell}$, while constraints (11) link completion
 342 time variables $S_i + p_i$ to variables $z_{f_i,\ell}$.

343 The remaining constraints allow to compute the intersection lengths $d_{i,\ell}$. Constraints (12) enforce $d_{i,\ell}$ to take
 344 value 0 when period ℓ is either before or after the execution interval of activity i , and value Δ when period ℓ is
 345 integrally included in the execution interval of i . Constraints (13) allow to compute $d_{i,\ell}$ when ℓ is the period that
 346 contains S_i , while $S_i + p_i$ belongs to a period $\ell' > \ell$. Constraints (14) allow to compute $d_{i,\ell}$ when ℓ is the period that
 347 contains $S_i + p_i$ while S_i belongs to a period $\ell' < \ell$. Constraints (15) state that the sum of the intersection lengths of
 348 activity i over all the periods must be equal to the processing time of i . These constraints are necessary to compute

349 the correct d_{ij} when the duration of an activity is lower than Δ and the activity is fully included in one period (see
 350 proof of Theorem 4 for further details).

351 **Theorem 4.** *Formulation (F1) is a correct formulation for the PARCPSP*

352 Proof is given in Appendix A.

353 4.1.3 Strengthening the first formulation

Morin et al. 2017 proposed to strengthen the formulation as follows. Since all periods have the same duration Δ , starting the project in the first period is a dominant policy. Hence the following constraint is valid.

$$0 \leq S_0 \leq \Delta \quad (16)$$

354 Furthermore, since preemption is not allowed, the number of periods intersected by an activity is bounded as
 355 stated by the following theorem. As in the proof of Theorem 4, let us define the first period of an activity ℓs^i as the
 356 one that satisfies $(\ell s^i - 1)\Delta \leq S_i < \ell s^i \Delta$ and let last period of an activity ℓf^i be defined by $(\ell f^i - 1)\Delta \leq S_i + p_i < \ell f^i \Delta$.

357 **Lemma 1.** *The first and the last period of an activity are such that either $\ell f^i = \ell s^i + \lfloor \frac{p_i}{\Delta} \rfloor$ or $\ell f^i = \ell s^i + \lceil \frac{p_i}{\Delta} \rceil$.*

Proof. Since we have $(\ell s^i - 1)\Delta \leq S_i < \ell s^i \Delta$, it follows:

$$\begin{aligned} (\ell s^i - 1)\Delta + p_i &\leq S_i + p_i < \ell s^i \Delta + p_i \\ \Leftrightarrow (\ell s^i - 1 + \frac{p_i}{\Delta})\Delta &\leq S_i + p_i < (\ell s^i + \frac{p_i}{\Delta})\Delta \\ \Rightarrow (\ell s^i - 1 + \lfloor \frac{p_i}{\Delta} \rfloor)\Delta &\leq S_i + p_i < (\ell s^i + \lceil \frac{p_i}{\Delta} \rceil)\Delta, \end{aligned}$$

358 which yields the desired result. □

359 In the proof of Theorem 4 (Appendix A), we show that for any solution S_i , $i \in \mathcal{A}$, a compatible assignment of
 360 the other variables can be obtained by setting $z_{S_i, \ell} = 0$ for each $\ell < \ell s^i$, $z_{S_i, \ell} = 1$ for each $\ell \geq \ell s^i$, $z_{f_i, \ell} = 0$ for each
 361 $\ell < \ell f^i$ and $z_{f_i, \ell} = 1$ for each $\ell \geq \ell f^i$.

362 Hence, a consequence of Lemma 1 is that variables $z_{S_i, \ell}$ and $z_{f_i, \ell}$ can be linked via a binary variable π_i (only one
 363 binary variable per activity), such that:

$$\begin{aligned} \pi_i = 0 &\Leftrightarrow z_{S_i, \ell} = z_{f_i, \ell + \lfloor \frac{p_i}{\Delta} \rfloor} \\ \pi_i = 1 &\Leftrightarrow z_{S_i, \ell} = z_{f_i, \ell + \lceil \frac{p_i}{\Delta} \rceil} \end{aligned}$$

365 In this case the integrality constraint on variables $z_{f_i, \ell}$ can be relaxed and the linking constraint can be easily
 366 linearized by the adjunction of the following constraints:

$$zf_{i,\ell} \in [0, 1] \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (17)$$

$$\pi_i \in \{0, 1\} \quad \forall i \in \mathcal{A} \quad (18)$$

$$zs_{i,\ell} \geq zf_{i,\ell + \lfloor \frac{p_i}{\Delta} \rfloor} \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (19)$$

$$zs_{i,\ell} \leq zf_{i,\ell + \lceil \frac{p_i}{\Delta} \rceil} \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (20)$$

$$zs_{i,\ell} \leq zf_{i,\ell + \lfloor \frac{p_i}{\Delta} \rfloor} + \pi_i \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (21)$$

$$zs_{i,\ell} \geq zf_{i,\ell + \lceil \frac{p_i}{\Delta} \rceil} + \pi_i - 1 \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (22)$$

367 If the number of periods is large this reduces considerably the number of explicit binary variables of the problem.

368 *Remark.* If activity $i \in \mathcal{A}$ is such that $p_i \bmod \Delta = 0$, the first and the last period of this activity are such that
 369 $\ell f^i = \ell s^i + \frac{p_i}{\Delta}$. Then there is no need to introduce variable π_i and the above-defined constraints can be simply
 370 replaced by:

$$zs_{i,\ell} = zf_{i,\ell + \frac{p_i}{\Delta}} \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (23)$$

371 We denote by $(F1s)$ the strengthened formulation of Morin et al. (2017).

372 4.2 An alternative formulation

373 4.2.1 Variables description

In time-indexed formulations of scheduling problems, precedence constraints expressed directly under the form of constraints (8) are called aggregated precedence constraints. There exists indeed a disaggregated form of these precedence constraints that strengthen the relaxation (see e.g. Artigues (2017)). We show in this section that a disaggregated form of the precedence constraints can be proposed for the PARCPSP despite the continuous nature of the start time variables. For each activity $i \in \mathcal{A}$ and each time period $\ell \in \mathcal{L}$, let us define new variables $\lambda_{i,\ell}$ and $\mu_{i,\ell}$ such that

$$\lambda_{i,\ell} = |[0, S_i] \cap [(\ell - 1)\Delta, \ell\Delta]| \text{ and } \mu_{i,\ell} = |[S_i + p_i, L\Delta] \cap [(\ell - 1)\Delta, \ell\Delta]|.$$

374 The other decision variables used in the new model are described in table 3.

S_i	Start time of activity $i \in \mathcal{A}$ S_0 (respectively S_{n+1}) represents the start (resp. the end) of the project.
$d_{i,\ell}$	intersection length of intervals $[S_i, S_i + p_i]$ and $[(\ell - 1)\Delta, \ell\Delta]$
$\lambda_{i,\ell}$	intersection length of intervals $[0, S_i]$ and $[(\ell - 1)\Delta, \ell\Delta]$
$\mu_{i,\ell}$	intersection length of intervals $[S_i + p_i, L\Delta]$ and $[(\ell - 1)\Delta, \ell\Delta]$
$z_{i,\ell}^\lambda$	Binary variables ensuring a decreasing step behavior for variables $\lambda_{i,\ell}$
$z_{i,\ell}^\mu$	Binary variables ensuring an increasing step behavior for variables $\mu_{i,\ell}$

Table 3: Variables of the second period-indexed formulation

375 From this definition it immediately follows that $\lambda_{i,\ell}$ is a decreasing step function of ℓ , while, symmetrically, $\mu_{i,\ell}$
376 is an increasing step function of ℓ . In the case that $p_i \geq \Delta$, illustrated by Figure 9 for activity A_i , $\lambda_{i,\ell}$ is equal to Δ
377 for each period $\ell < \ell s^i$, then equal to $\Delta - d_{i,\ell}$ for $\ell = \ell s^i$ and finally equal to 0 for $\ell > \ell s^i$. Under the same condition
378 ($p_i \geq \Delta$), $\mu_{i,\ell}$ is equal to 0 for $\ell < \ell f^i$, then equal to $\Delta - d_{i,\ell}$ for $\ell = \ell f^i$ and finally equal to Δ for $\ell > \ell f^i$.

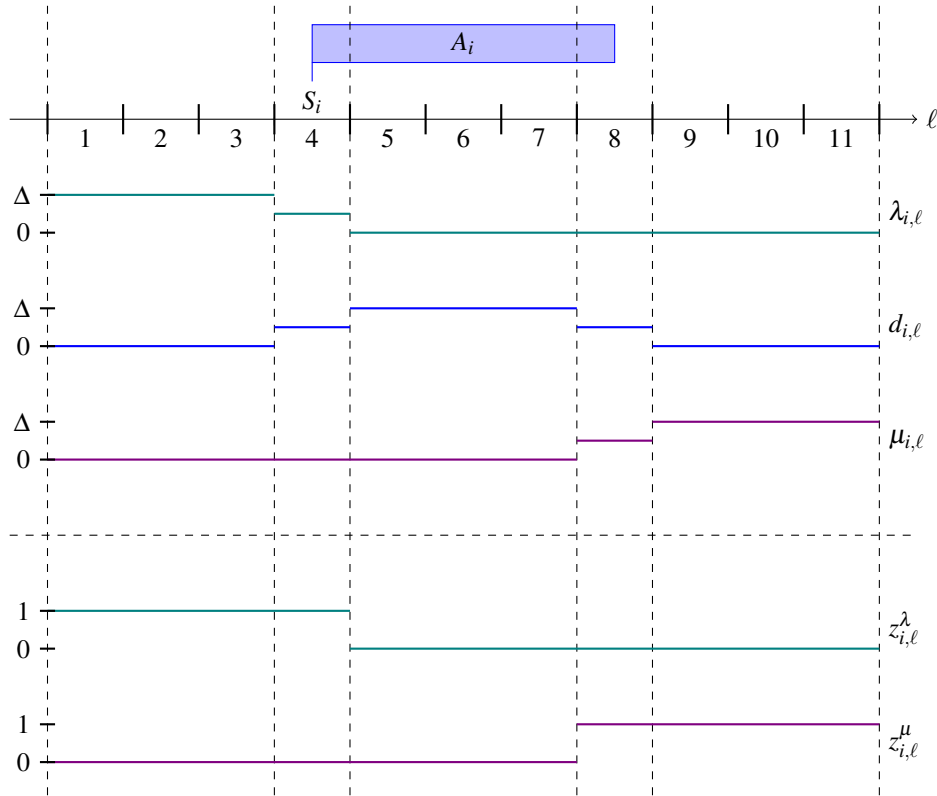


Figure 9: Scheduling variables of an activity for the second period-indexed formulation with $p_i \geq \Delta$

379 In the case where $p_i < \Delta$ and if the execution of activity i overlaps a period change (precisely $\ell s^i \Delta \in [S_i, S_i + p_i]$)
380 the same behavior is observed.

381 In the case where $p_i < \Delta$ and there is no period ℓ such that $\ell\Delta \in [S_i, S_i + p_i]$ (such as for Activity A_i , fully
382 included in period 2 in Figure 10), then a slightly different behavior is observed. The difference in this case is that
383 for $\ell = \ell s^i = \ell c^i$, the period that fully includes the activity, we have $\lambda_{i,\ell} + d_{i,\ell} + \mu_{i,\ell} = \Delta$ with $\lambda_{i,\ell} > 0$ and $\mu_{i,\ell} > 0$.
384 In Figure 10, we have $\lambda_{i,2} = 0.5\Delta$ and $\mu_{i,2} = d_{i,2} = 0.25\Delta$.

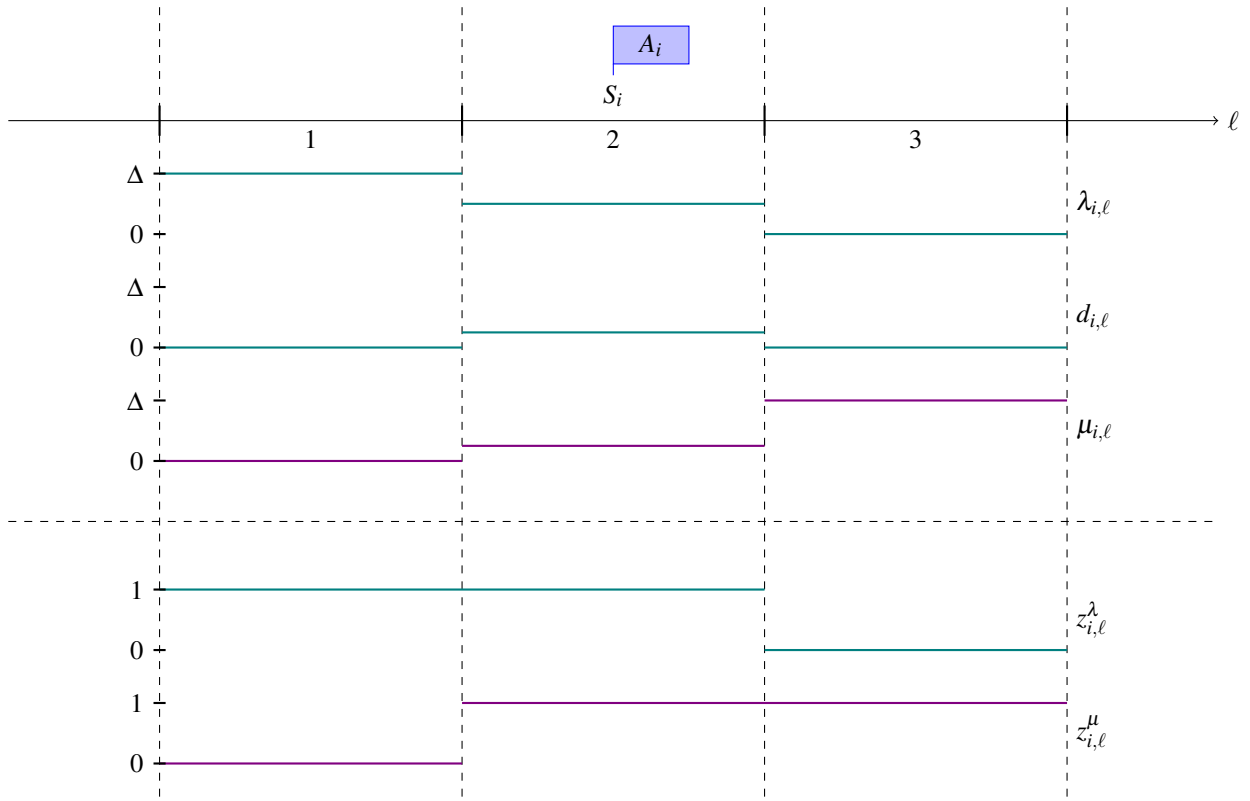


Figure 10: Scheduling variables of an activity for the second period-indexed formulation with $p_i < \Delta$

385 The monotonicity of the new variables allow a simpler linearization and furthermore, by definition, $\lambda_{i,\ell}$, $\mu_{i,\ell}$ and
 386 $d_{i,\ell}$ define a partition of period ℓ . More precisely, we always have $\lambda_{i,\ell} + d_{i,\ell} + \mu_{i,\ell} = \Delta$.

Given the proposed variables, the new formulation can be written as follows.

$$(F2) \quad \text{Minimize} \quad S_{n+1} - S_0 \quad (24)$$

$$S_{i_2} - S_{i_1} \geq p_{i_1} \quad \forall (i_1, i_2) \in E \quad (25)$$

$$\sum_{i \in \mathcal{A}} r_{i,k} d_{i,\ell} \leq b_k \Delta \quad \forall k \in \mathcal{R}, \forall \ell \in \mathcal{L} \quad (26)$$

$$\lambda_{i,\ell} + d_{i,\ell} + \mu_{i,\ell} = \Delta \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (27)$$

$$S_i = \sum_{\ell \in \mathcal{L}} \lambda_{i,\ell} \quad \forall i \in \mathcal{A} \quad (28)$$

$$\sum_{\ell \in \mathcal{L}} d_{i,\ell} = p_i \quad \forall i \in \mathcal{A} \quad (29)$$

$$\lambda_{i,\ell} \leq \Delta z_{i,\ell}^\lambda \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (30)$$

$$\lambda_{i,\ell} \geq \Delta z_{i,\ell+1}^\lambda \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (31)$$

$$\mu_{i,\ell} \leq \Delta z_{i,\ell}^\mu \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (32)$$

$$\mu_{i,\ell} \geq \Delta z_{i,\ell-1}^\mu \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (33)$$

$$z_{i,\ell}^\lambda \in \{0, 1\} \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (34)$$

$$z_{i,\ell}^\mu \in \{0, 1\} \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (35)$$

$$d_{i,\ell}, \mu_{i,\ell}, \lambda_{i,\ell} \geq 0 \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (36)$$

Objective function (24), precedence constraints (25) and resource constraints (26) are the same as in the first formulation. Constraints (27) define the partition of each period ℓ by variables $\lambda_{i,\ell}$, $\mu_{i,\ell}$ and $d_{i,\ell}$. Constraints (28) allow to express S_i from the $\lambda_{i,\ell}$ variables. Constraints (29) take the activity processing times into account. Constraints (30) and (31) define the step behavior of variables $\lambda_{i,\ell}$ and $z_{i,\ell}^\lambda$, in such a way that a single variable $\lambda_{i,\ell}$ may vary between 0 and Δ , while the others take either value 0 or value Δ . Constraints (32) and (33) define the same process for variables $\mu_{i,\ell}$ and $z_{i,\ell}^\mu$ (cf figure 9). Finally $z_{i,\ell}^\lambda$ and $z_{i,\ell}^\mu$ are binary variables (constraints 34 and 35) while $d_{i,\ell}$, $\mu_{i,\ell}$ and $\lambda_{i,\ell}$ are non negative (constraints 36).

Theorem 5. *Formulation (F2) is a correct formulation of the PARCPSP.*

Proof is given in Appendix B.

4.2.3 Formulation strengthening

As for the previous formulation the start time of the project can be assigned to the first period.

$$0 \leq S_0 \leq \Delta \quad (37)$$

As a consequence of Lemma 1, a binary variable π_i can also be defined for each activity to express the link between the start and the first period of an activity

$$\pi_i = 0 \Leftrightarrow z_{i,\ell}^\lambda + z_{i,\ell + \lfloor \frac{p_i}{\Delta} \rfloor - 1}^\mu = 1$$

$$\pi_i = 1 \Leftrightarrow z_{i,\ell}^\lambda + z_{i,\ell + \lceil \frac{p_i}{\Delta} \rceil - 1}^\mu = 1$$

The linearization of these constraints gives:

$$z_{i,\ell}^\mu \in [0, 1] \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (38)$$

$$\pi_i \in \{0, 1\} \quad \forall i \in \mathcal{A} \quad (39)$$

$$z_{i,\ell}^\lambda + z_{i,\ell + \lfloor \frac{p_i}{\Delta} \rfloor - 1}^\mu \leq 1 \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (40)$$

$$z_{i,\ell}^\lambda + z_{i,\ell + \lceil \frac{p_i}{\Delta} \rceil - 1}^\mu \geq 1 \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (41)$$

$$z_{i,\ell}^\lambda + z_{i,\ell + \lfloor \frac{p_i}{\Delta} \rfloor - 1}^\mu \geq 1 - \pi_i \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (42)$$

$$z_{i,\ell}^\lambda + z_{i,\ell + \lceil \frac{p_i}{\Delta} \rceil - 1}^\mu \leq 2 - \pi_i \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (43)$$

401 *Remark.* As for (F1) if an activity $i \in \mathcal{A}$ is such that $p_i \bmod \Delta = 0$, there is no need to introduce π_i for this activity,
402 as the last period of the activity can be obtained by a constant translation from the first period.

$$z_{i,\ell}^\lambda + z_{i,\ell + \frac{p_i}{\Delta} - 1}^\mu = 1 \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (44)$$

$$\lambda_{i,\ell} + \mu_{i,\ell + \frac{p_i}{\Delta}} = \Delta \quad \forall i \in \mathcal{A}, \forall \ell \in \mathcal{L} \quad (45)$$

403 We denote by (F2s) the so-strengthened formulation.

404 4.2.4 Disaggregated precedence constraints

405 Consider two formulations A and B , and let $z_A(I)$ ($z_B(I)$) denote the value of the linear relaxation of model A
406 (B) applied to instance I of the PARCPSP. Following standard terminology, we say that the linear relaxation of
407 Formulation A is stronger than the linear relaxation of Formulation B when the two following conditions are fulfilled:

408 C1: for each instance I of PARCPSP, $z_A(I) \geq z_B(I)$, and

409 C2: there exists an instance I of PARCPSP for which $z_A(I) > z_B(I)$.

410 Thanks to the introduction of variables $\lambda_{i,\ell}$ and $\mu_{i,\ell}$, a further tightening of the formulation (F2s) can be obtained
411 through the definition of disaggregated precedence constraints. For that purpose, aggregated precedence constraints
412 (25) can be replaced by:

$$\mu_{i_1,\ell} + \lambda_{i_2,\ell} \geq \Delta \quad \forall (i_1, i_2) \in E, \forall \ell \in \mathcal{L} \quad (46)$$

413 **Theorem 6.** Replacing in formulation (F2s), aggregated precedence constraints (25) by disaggregated constraints
414 (46) yields a correct formulation for the PARCPSP, which is stronger.

Proof. It is easy to see that the aggregated precedence constraints (25) are implied by the conjunction of disaggregated constraints (46) and constraints (27–29). Indeed, summing up constraints (46) for all $l \in \mathcal{L}$ yields:

$$\sum_{l \in \mathcal{L}} \lambda_{i_2,\ell} \geq L\Delta - \sum_{l \in \mathcal{L}} \mu_{i_1,\ell}$$

This gives the aggregated precedence constraints since $S_i = \sum_{l \in \mathcal{L}} \lambda_{i,\ell}$ by (28) and $S_i + p_i = L\Delta - \sum_{l \in \mathcal{L}} \mu_{i,\ell}$ by (27–29). Hence we have shown that the LP relaxation of the new formulation with the disaggregated precedence constraints is not weaker than the new formulation with the aggregated precedence constraints. Consider now the problem instance with $L = 3$ periods of duration $\Delta = 1$ and $n = 3$ activities with durations $p_1 = p_2 = p_3 = 1$ and a single resource of capacity $b_1 = 3$ and activity requirements $b_1 = b_2 = 2$ and $b_3 = 3$. Furthermore there are two precedence constraints $E = \{(1,3), (2,3)\}$. Consider the following (optimal) fractional solution of (F2s), with objective value 2.

	S_i	$\lambda_{i,1}$	$\lambda_{i,2}$	$\lambda_{i,3}$	$z_{i,1}^\lambda$	$z_{i,2}^\lambda$	$z_{i,3}^\lambda$	$\mu_{i,1}$	$\mu_{i,2}$	$\mu_{i,3}$	$z_{i,1}^\mu$	$z_{i,2}^\mu$	$z_{i,3}^\mu$	$d_{i,1}$	$d_{i,2}$	$d_{i,3}$
1	$\frac{3}{4}$	$\frac{3}{4}$	0	0	1	0	0	0	$\frac{1}{4}$	1	0	1	1	$\frac{1}{4}$	$\frac{3}{4}$	0
2	$\frac{3}{4}$	$\frac{3}{4}$	0	0	1	0	0	0	$\frac{1}{4}$	1	0	1	1	$\frac{1}{4}$	$\frac{3}{4}$	0
3	$\frac{7}{4}$	$\frac{7}{8}$	$\frac{7}{8}$	0	1	$\frac{7}{8}$	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0	$\frac{1}{8}$	1	$\frac{1}{8}$	0	$\frac{7}{8}$

This solution satisfies the LP relaxation of constraints (25–36) but violates the disaggregated constraints. For period $\ell = 1$ and precedence (1,3), we have $\mu_{1,1} + \lambda_{3,1} = \frac{7}{8} < \Delta$ although we have $S_3 = \frac{7}{4} \geq S_2 + p_2 = \frac{7}{4}$. Hence the new formulation augmented with the disaggregated precedence constraints is stronger. Furthermore solving the LP relaxation with the disaggregated constraint gives the following solution with optimal solution $\frac{25}{12} > 2$. Furthermore, since the $z_{i,\ell}^\lambda$ and $z_{i,\ell}^\mu$ variables are all integer-valued, the solution of the relaxation is feasible for the PARCPSP and consequently optimal, which illustrates the potential quality of the new valid inequalities.

	S_i	$\lambda_{i,1}$	$\lambda_{i,2}$	$\lambda_{i,3}$	$z_{i,1}^\lambda$	$z_{i,2}^\lambda$	$z_{i,3}^\lambda$	$\mu_{i,1}$	$\mu_{i,2}$	$\mu_{i,3}$	$z_{i,1}^\mu$	$z_{i,2}^\mu$	$z_{i,3}^\mu$	$d_{i,1}$	$d_{i,2}$	$d_{i,3}$
1	$\frac{1}{4}$	$\frac{1}{4}$	0	0	1	0	0	0	$\frac{3}{4}$	1	0	1	1	$\frac{3}{4}$	$\frac{1}{4}$	0
2	$\frac{1}{4}$	$\frac{1}{4}$	0	0	1	0	0	0	$\frac{3}{4}$	1	0	1	1	$\frac{3}{4}$	$\frac{1}{4}$	0
3	$\frac{4}{3}$	1	$\frac{1}{3}$	0	1	1	0	0	0	$\frac{2}{3}$	0	0	1	0	$\frac{2}{3}$	$\frac{1}{3}$

415

□

416 We denote by (F2s+) the new formulation with the disaggregated precedence constraints.

417 4.3 Theoretical comparison of formulations (F1s) and (F2s+)

418 **Theorem 7.** *The linear relaxation of (F2s+) is stronger than the linear relaxation of (F1s).*

Proof. Let us first compare the relaxations of formulations (F1s) and (F2s), with aggregated precedence constraints only. We remark there exist linear non singular transformations between the binary variables ($zs_{i,\ell}$ and $zf_{i,\ell}$) of the first model and the one of the second model ($z_{i,\ell}^\lambda$ and $z_{i,\ell}^\mu$).

$$zs_{i,\ell} = 1 - z_{i,\ell+1}^\lambda$$

$$zf_{i,\ell} = z_{i,\ell}^\mu$$

419 Continuous variables (S_i and $d_{i,\ell}$) appear in both models with the same meaning, while variables ($\lambda_{i,\ell}$ and $\mu_{i,\ell}$)
420 appear only in the second model.

The start time of an activity i is a linear expression of variables $\lambda_{i,\ell}$ (Constraints (28)).

$$S_i = \sum_{\ell=1}^L \lambda_{i,\ell}$$

Similarly, recall that the completion time of an activity i is a linear expression of variables $\mu_{i,\ell}$ (using constraints (27) to (29)).

$$S_i + p_i = L\Delta - \sum_{\ell=1}^L \mu_{i,\ell}$$

Aggregated precedence constraints have the same expression in both models (constraints (8) and (25)). We remark that rewriting the other constraints of formulation ($F1s$) by substituting variables of the first model by the variables of the second model yields constraints that are implied by the constraints of ($F2s$). Let us provide the proof for the lower bound part of Constraints (10). We first rewrite the constraint for activity i and period $\ell - 1$, by using the transformation $z_{S_{i,\ell-1}} = 1 - z_{i,\ell}^\lambda$, we obtain the following equivalent constraint in variable $z_{i,\ell}^\lambda$.

$$S_i \geq (\ell - 1)\Delta - (\ell - 1)\Delta \left(1 - z_{i,\ell}^\lambda\right) \quad (10'_{LB})$$

Now we evaluate expression $S_i - (\ell - 1)\Delta + (\ell - 1)\Delta \left(1 - z_{i,\ell}^\lambda\right)$ by using $S_i = \sum_{\ell=1}^L \lambda_{i,\ell}$. We obtain:

$$\begin{aligned} & S_i - (\ell - 1)\Delta + (\ell - 1)\Delta \left(1 - z_{i,\ell}^\lambda\right) \\ &= -(\ell - 1)\Delta + \left(\sum_{\ell'=1}^L \lambda_{i,\ell'}\right) + (\ell - 1)\Delta - \left(\sum_{\ell'=1}^{\ell-1} \Delta\right) z_{i,\ell}^\lambda \\ &\geq \sum_{\ell'=1}^L \lambda_{i,\ell'} - \sum_{\ell'=1}^{\ell-1} \Delta z_{i,\ell'+1}^\lambda \\ &\geq \sum_{\ell'=1}^L \lambda_{i,\ell'} - \sum_{\ell'=1}^{\ell-1} \lambda_{i,\ell'} \\ &= \sum_{\ell'=\ell}^L \lambda_{i,\ell'} \\ &\geq 0 \end{aligned}$$

421 The proof for the upper bound part of Constraints (10) and Constraints (11) (link between S_i , $z_{S_{i,\ell}}$ and $z_{f_{i,\ell}}$) and
 422 the proof for Constraints (12) to (14) (expression of $d_{i,\ell}$) are given in the Appendix. Constraints (15) of the first
 423 model are also present in the second model (Constraints (36) and (29)).

424 Lastly, Constraints (19) to (23) that link variables $z_{S_{i,\ell}}$ and $z_{f_{i,\ell}}$ of the first model via binary variable π_i are
 425 equivalent to constraints (40) to (44) that link variables $z_{i,\ell}^\lambda$ and $z_{i,\ell}^\mu$ of the second model via the same binary variable
 426 π_i : The above described linear transformations can be used to switch from one formulation to the other. From
 427 what precedes, we conclude that formulation ($F2s$) cannot be weaker than formulation ($F1s$) in terms of linear
 428 programming relaxation. As Theorem 6 states that formulation ($F2s+$) is stronger than formulation ($F2s$), the
 429 result follows. \square

430 5 Computational experiments

431 In this section, we compare the different MILP formulations on a set of benchmark instances from the literature. As
 432 in Morin et al. (2017), we select standard resource-constrained-project scheduling instances, to which we associate
 433 a period Δ with $\Delta = 1, 2, 3, 4$ and 5 . We use IBM ILOG CPLEX 20.1 for solving the (mixed-integer) linear programs
 434 with default parameters. All experiments were run with 2 threads on 8 cluster nodes, each with 36 Intel Xeon CPU
 435 E5-2695 v3 2.10GHz cores running Linux Ubuntu 16.04.4.

436 We first compare the LP relaxations of the Morin et al. (2017) strengthened formulation ($F1s$) with the new
 437 ones ($F2s$ and $F2s+$) on the 30, 60, 90 and 120 activity RCPSp instances from the PSPLIB library, named KSD30,
 438 KSD60, KSD90 and KSD120 (Kolisch and Sprecher 1996), as well as on the Pack instances (Carlier and Néron

2003). With the different values of Δ , we obtain a set of 2400 KSD30 instances, 2400 KSD60 instances, 2400 KSD90 instances, 3000 KSD120 instances and 280 Pack instances. For each instance, an upper bound of the number of periods is obtained by selecting the best solution in terms of project duration returned by the randomized multi-start priority-rule based heuristic presented in Morin et al. (2017) with 1000 iterations.

Table 4 reports the results of the LP relaxations compared to the trivial critical path lower bound (CPM) given by the precedence constraints only. Each row of the table correspond to the instances of a specific set for a given Δ , except the last row “all” of each instance set that regroups all the Δ values and the last “all” row that regroups the statistics over all instances and Δ values. Column #UB>CPM displays the number of instances for which the upper bound is not equal to the CPM lower bound. Indeed, the LP relaxations have the potential of increasing the CPM lower bound only on these instances. A first remark is that this number is a decreasing function of Δ , which illustrates the fact that decreasing the period lengths globally tightens the resource constraints, yielding larger project durations. There are two columns of results for each formulation $F1s$, $F2s$ and $F2s+$. The first column (gap CPM) gives the average improvement upon the CPM bound, only on the instances for which the CPM bound is strictly lower than UB (number given in column #UB>CPM). The second column (cpu) gives the average cpu time in seconds. The best results in terms of gap and CPU time are highlighted in bold.

The ranking $F1s < F2s < F2s+$ from the weakest to the strongest upper bound is well illustrated by the results. Globally, for the larger values of Δ , all bounds are rather weak and each bound gets tighter as Δ decreases. The improvement brought by the new formulation with aggregated precedence constraints ($F2s$) on the previous formulation is modest, except on the Pack set. The new formulation with disaggregated precedence constraints ($F2s+$) significantly improves the previous formulation upon the CPM-based lower bound on all instances with small Δ . The large gaps observed for the Pack instance set are explained by the small number of precedence constraints in this set and the predominance of resource constraints. This allows to remark that the improvement brought by $F2s$ and $F2s+$ on the previous formulation $F1s$ can be drastic and indicates that the new formulation better captures resource conflicts. About the computational times, the fastest bounds are obtained either by $F1s$ or $F2s$, the latter offering the best compromise quality/speed. The computational times become very large for the KSD120 set and illustrates the limits of time-indexed MILP approaches for large scheduling horizons, even with aggregated resource constraints.

The $F2s+$ formulation is superior to the other ones in terms of LP relaxations. We now switch to the comparison of the quality of the integer solutions found by CPLEX under a limited time.

We limit the CPU time to 1 hour for the KSD30 instances and to 2 hours for the remaining instances. The randomized multi-start priority-rule based heuristic of Morin et al. (2017) with 1000 iterations is used to obtain an initial feasible solution provided as a “MILP start”.

Table 5 reports the obtained results on the KSD30 and KSD60 sets for the three formulations. For each formulation and value of Δ , the table displays the number of optimal solutions found and certified within the allotted time (column #opt), the average gap between the lower and the upper bound returned by the solver, and the average CPU time. The last column (av. gap LBRCPS) gives the average gap for each value of Δ of the optimal solution (or the best found lower bound when optimality is not verified) for the PARCPSP to the optimal solution (of the best known lower bound when the optimum is unknown) of the RCPS. The number in this column for row all is the

set	Δ	#UB>CPM	$F1s$		$F2s$		$F2s+$	
			gap CPM	cpu (s)	gap CPM	cpu (s)	gap CPM	cpu (s)
KSD30	5	146	0.00%	0.11	0.00%	0.23	0.23%	0.30
	4	171	0.00%	0.15	0.00%	0.27	0.21%	0.38
	3	198	0.00%	0.20	0.02%	0.33	0.85%	0.41
	2	234	0.00%	0.31	0.20%	0.47	1.89%	0.41
	1	264	0.02%	0.54	1.00%	0.62	4.00%	0.33
	all	1013	0.01%	0.29	0.31%	0.41	1.71%	0.37
KSD60	5	158	0.00%	0.51	0.00%	0.76	0.33%	2.02
	4	167	0.00%	0.91	0.00%	1.01	0.42%	2.93
	3	181	0.00%	1.40	0.04%	1.52	1.51%	3.29
	2	202	0.00%	3.32	0.49%	2.56	3.16%	4.57
	1	233	0.00%	6.97	2.05%	3.19	5.73%	7.76
	all	941	0.00%	2.96	0.62%	1.94	2.52%	4.39
KSD90	5	164	0.00%	1.68	0.00%	2.88	0.27%	5.14
	4	174	0.00%	3.82	0.00%	2.96	0.34%	6.64
	3	179	0.00%	8.11	0.01%	5.16	1.39%	19.06
	2	198	0.00%	28.82	0.52%	13.86	3.60%	37.46
	1	214	0.00%	57.28	2.43%	22.01	7.12%	46.93
	all	929	0.00%	21.91	0.67%	10.08	2.79%	24.62
KSD120	5	485	0.00%	10.71	0.07%	5.03	1.48%	14.43
	4	496	0.00%	27.98	0.22%	8.45	1.84%	31.09
	3	508	0.00%	72.78	1.07%	17.56	3.75%	66.96
	2	521	0.00%	145.90	2.90%	48.35	6.83%	138.41
	1	550	0.01%	282.09	6.22%	73.84	11.52%	247.16
	all	2560	0.00%	112.19	2.19%	31.78	5.25%	103.31
Pack	5	54	0.00%	0.09	1.74%	0.13	15.72%	0.19
	4	55	0.00%	0.13	3.97%	0.19	20.37%	0.22
	3	55	0.00%	0.26	14.05%	0.26	35.64%	0.30
	2	55	0.31%	0.51	32.26%	0.37	51.32%	0.41
	1	55	4.45%	1.28	64.17%	0.46	73.89%	0.62
	all	274	0.96%	0.46	23.32%	0.28	39.48%	0.35
all	all	5717	0.00%	54.34	1.25%	16.26	3.52%	51.05

Table 4: Comparisons of $F1s$, $F2s$ and $F2s+$ LP relaxations on various instance sets

best gap over all Δ values. Note that for KSD30 instances the optimal makespan for the RCPSP are known while for KSD60 we use for comparison the best current LB¹.

The av. gap LB RCPSP gaps is increasing on average in function of Δ and is of significant magnitude. This confirms that aggregating the resource constraints without restricting the start time values is highly beneficial for reducing the makespan, as mentioned in the introduction. For each instance set, the largest obtained bound for the different values of Δ gives a gap to the best known RCPSP LB of less than 3%. No lower bound is improved on the KSD60 set. The results in table 5, compared to the best results obtained by MILP for the standard RCPSP in Koné et al. (2011), suggest that the PARCPSP is not much easier to solve than the RCPSP. So it is still unclear whether the PARCPSP can be used as an efficient bounding scheme of the RCPSP.

Turning now to the comparison of formulations, the best results are displayed in bold. The aggregated variant of the new formulation ($F2s$) appears dominated on all criteria, including the CPU time². The previous formulation ($F1s$) obtains the best results for solving the KSD30 instances with $\Delta = 1$ and $\Delta = 2$ as well as the KSD60 instances with $\Delta = 3$ in terms of optimal solutions found with a faster or equivalent CPU time. This indicates that the quality of the LP relaxation of $F2s+$ does not always compensate the search slowdown it incurs. However, the $F2s+$ dominates on all criteria for the remaining instances and is always the best one in terms of average gap for all Δ values. Averaging all instances and all Δ values, the $F2s+$ formulation outperforms the other ones for all criteria.

set	Δ	$F1s$			$F2s$			$F2s+$			gap LB RCPSP
		#opt	gap	time	#opt	gap	time	#opt	gap	time	
KSD30	5	457	0.13%	226.40	447	0.24%	323.09	474	0.02%	110.41	6.70%
	4	441	0.39%	342.46	432	0.52%	411.52	459	0.14%	234.62	6.25%
	3	438	0.62%	386.32	423	0.89%	465.93	437	0.42%	380.21	5.69%
	2	436	0.97%	386.46	425	1.41%	477.13	433	0.83%	430.54	4.72%
	1	447	1.05%	336.47	422	2.04%	492.52	437	1.03%	406.05	2.82%
	all	2219	0.63%	335.62	2149	1.02%	434.04	2240	0.49%	312.36	2.82%
KSD60	5	402	0.97%	1235.91	400	1.22%	1290.65	426	0.32%	974.67	3.57%
	4	397	1.36%	1278.81	394	1.89%	1337.82	406	0.79%	1224.08	3.48%
	3	395	2.08%	1335.99	390	3.64%	1388.57	394	1.57%	1354.49	3.35%
	2	383	3.37%	1504.17	381	5.86%	1534.21	383	2.77%	1498.22	3.17%
	1	377	6.98%	1604.12	372	6.91%	1664.11	380	3.59%	1535.03	2.50%
	all	1954	2.95%	1391.80	1937	3.90%	1443.07	1989	1.81%	1317.30	2.45%

Table 5: Comparisons of integer solutions for instances KSD30 and KSD60

We now switch to the KSD90, KSD120 and Pack benchmarks, which are much harder to solve in the RCPSP setting. Here, only the non dominated formulations ($F1s$ and $F2s+$) are compared. As seen in Table 6, except for three exceptions (average CPU time criterion for KSD90- $\Delta = 3$ instances and number of optima found for KSD90- $\Delta = 2$ instances), the new formulation outperforms the previous one on all instances and all criteria. Two additional observations are worth mentioning. First while the optimality gaps moderately increase for KSD30, KSD60 and

¹Recorded at <http://solutionsupdate.ugent.be/>, last visit Novembre 9, 2021

²Recall that the time limit is 1 hour for KSD30 and 2 hours for KSD60

498 KSD90 sets, the limit of the time-indexed MILP approach seems to be reached for the KSD120 set since large gaps
499 are observed as Δ decreases. This is inline with the large needed CPU time for solving the LP relaxation. A second
500 remark is the relative quality of the RCPSP bbound on the Pack instances. These instances seem as challenging in
501 the PARCPSP setting as they are in the RCPSP setting, even for $\Delta = 5$ instances since only 38 instances out of 55 are
502 solved to optimality. However 4 of the RCPSP lower bounds reported in Schutt et al. (2013) were improved, while
503 no lower bound was improved for KSD60, KSD90 and KSD120 instances. The main notorious difference between
504 the Pack and the KSD sets is that the Pack instances are “highly cumulative” in the sense that many activities can
505 be scheduled in parallel and have very few precedence constraints (which explains the name Pack with reference to
506 the 2D packing problem). In this case, the resource aggregation seems to pay off although all improvements were
507 obtained for $\Delta = 1$. The improved lower bounds are reported in Table 7.

set	Δ	$F1s$			$F2s+$			gap LB RCPSP
		#opt	gap	time	#opt	gap	time	
KSD90	5	390	1.55%	1366.66	411	0.86%	1208.99	1.88%
	4	388	2.25%	1401.56	393	1.85%	1334.47	2.01%
	3	385	3.36%	1439.20	387	2.89%	1441.82	2.14%
	2	384	6.79%	1490.47	378	4.16%	1575.43	2.33%
	1	376	10.69%	1655.45	378	4.64%	1561.48	1.78%
	all	1923	4.93%	1470.67	1947	2.88%	1424.44	1.42%
KSD120	5	302	8.82%	3738.88	353	7.66%	3270.11	5.81%
	4	283	13.92%	3946.06	314	10.34%	3731.16	6.83%
	3	260	20.30%	4232.44	272	12.74%	4137.12	7.25%
	2	232	28.12%	4555.50	238	16.79%	4420.86	7.76%
	1	203	35.02%	4958.74	223	18.05%	4601.85	6.47%
	all	1280	21.24%	4286.32	1400	13.12%	4032.22	4.82%
Pack	5	27	1.12%	4937.18	38	2.02%	2959.33	11.77%
	4	19	2.04%	5577.67	37	0.47%	3292.10	9.24%
	3	16	3.27%	5787.41	27	3.10%	4179.22	7.44%
	2	13	5.08%	6185.74	26	2.24%	4632.17	5.01%
	1	6	8.26%	7085.33	32	1.89%	3857.74	1.85%
	all	63	3.95%	5914.67	160	1.94%	3784.11	1.83%

Table 6: Comparisons of integer solutions for instances KSD90, KSD120, Pack

name	LB (Schutt et al. 2013)	LB $F2s+$
Pack037	116	125
Pack046	110	118
Pack050	94	100
Pack053	97	105

Table 7: Improved RCPSP lower bounds on the Pack instance set compared to Schutt et al. (2013)

508 **6 Conclusion and perspectives**

509 In this paper, an original variant of the RCPSP, namely the PARCPSP, has been studied from a theoretical point
510 of view. This problem is indeed a relaxation of the RCPSP, that permits to model periodically aggregated resource
511 constraints arising from practical applications, where the resource usage is limited only on average over periods
512 of parameterized length. Contrarily to the RCPSP, the feasibility of a solution (with respect to the resource con-
513 straints) is no more invariant by shifting. We proposed three reductions to establish the computational complexity of
514 particular cases of the problem, which is strongly NP-hard in the general case. We designed a new period-indexed
515 mixed-integer linear programming formulation of the problem, defining the precedence constraint in a disaggregated
516 form. We carried out a polyhedral study that established that the new formulation is stronger than the previously
517 proposed formulation in terms of linear programming relaxation. A computational experiment on the set of PSPLIB
518 project scheduling instances with five different period lengths, showed that the practical improvement of the lower
519 bound is significant. When using the formulations for exact solution approaches in a commercial MILP solver, the
520 new formulation is still globally better in terms of optimal solution found and optimality gaps, except for a few
521 exceptions. The PARCPSP appears as a challenging NP-hard problem. Although it provides a bounding scheme
522 for the widely studied RCPSP, it is still unclear whether efficient approaches can be designed to this aim. However
523 this research direction is worth pursuing as a few lower bounds were improved for the difficult RCPSP instance
524 set Pack. For a global improvement of mixed-integer linear programming approaches, the disaggregated precedence
525 constraints could be added on-the-fly to obtain a better compromise between the formulation size and the relaxation
526 quality. The question whether an extended formulation based on a Dantzig-Wolfe decomposition of the resource
527 constraint, as successfully done for the RCPSP (Mingozzi et al. 1998; Brucker and Knust 2000; Baptiste and De-
528 massey 2004), would yield a competitive relaxation is open, as the aggregated resource constraints are less tight
529 than the standard ones. In order to fit practical applications, various extensions can be considered. For instance, the
530 definition of a consumption rate, either fixed (data) or variable (decision to make), on resources for each activity
531 would allow to model a wider range of resource usage profiles. Also, one could take into account additional limi-
532 tations, in a similar way as in Okubo et al. 2015, where a RCPSP/PI original formulation is enriched with specific
533 constraints. More flexible activities with variables intensities should also be considered such as in Hans 2001; Kis
534 2005. A promising research direction consists in considering varying period lengths. Indeed, models with time
535 buckets of non homogeneous lengths were successfully applied to a scheduling problem issued from particle ther-
536 apy for cancer treatment (Riedler et al. 2020). The latter work reveals that this approach has a double potential:
537 to better model practical situations where resource scarceness is time-dependent, and to improve primal and dual
538 bound for the RCPSP.

539 **Acknowledgement**

540 The research of Christian Artigues and Alain Haït is supported by ANR Project PER4MANCE (ANR-18-CE10-
541 0007). The research of Christian Artigues is partially supported by ANITI (ANR-19-PI3A-0004). The research of
542 Frits Spieksma is supported by NWO Gravitation Project NETWORKS, Grant Number 024.002.003. The research
543 of Tamás Kis was supported by the National Research, Development and Innovation Office – NKFIH, Grant no.

545 **References**

- 546 Artigues, C. (2017). “On the strength of time-indexed formulations for the resource-constrained project scheduling
547 problem”. In: *Operations Research Letters* 45.2, pp. 154–159.
- 548 Artigues, C., Gendreau, M., Rousseau, L.-M., and Vergnaud, A. (2009). “Solving an Integrated Employee
549 Timetabling and Job-Shop Scheduling Problem via Hybrid Branch-and-Bound”. In: *Computers & Operations
550 Research* 36.8, pp. 2330–2340.
- 551 Baptiste, P. and Demasse, S. (2004). “Tight LP bounds for resource constrained project scheduling”. In: *OR Spec-
552 trum* 26.2, pp. 251–262.
- 553 Blazewicz, J., Lenstra, J., and Rinnooy Kan, A. (1983). “Scheduling Subject to Resource Constraints: Classification
554 and Complexity”. In: *Discrete Applied Mathematics* 5.1, pp. 11–24.
- 555 Böttcher, J., Drexl, A., Kolisch, R., and Salewski, F. (1999). “Project scheduling under partially renewable resource
556 constraints”. In: *Management Science* 45.4, pp. 543–559.
- 557 Brucker, P. and Knust, S. (2000). “A linear programming and constraint propagation-based lower bound for the
558 RCPSP”. In: *European Journal of Operational Research* 127.2, pp. 355–362.
- 559 Carlier, J. and Néron, E. (2003). “On linear lower bounds for the resource constrained project scheduling problem”.
560 In: *European Journal of Operational Research* 149.2, pp. 314–324.
- 561 Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*.
562 W. H. Freeman.
- 563 Hait, A. and Artigues, C. (2011). “A Hybrid CP/MILP Method for Scheduling with Energy Costs”. In: *European
564 Journal of Industrial Engineering* 5.4, pp. 471–489.
- 565 Hans, E. W. (2001). “Resource Loading by Branch-and-Price Techniques”. Enschede: Twente Univ. Press.
- 566 Karp, R. M. (1972). “Reducibility Among Combinatorial Problems”. In: *Proceedings of a symposium on the Com-
567 plexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center,
568 Yorktown Heights, New York, USA*. Ed. by R. E. Miller and J. W. Thatcher. The IBM Research Symposia Series.
569 Plenum Press, New York, pp. 85–103.
- 570 Kis, T. (2005). “A Branch-and-Cut Algorithm for Scheduling of Projects with Variable-Intensity Activities”. In:
571 *Mathematical Programming* 103.3, pp. 515–539.
- 572 Kolisch, R. and Sprecher, A. (1996). “PSPLIB - A project scheduling library”. In: *European Journal of Operational
573 Research* 96, pp. 205–216.
- 574 Koné, O., Artigues, C., Lopez, P., and Mongeau, M. (2011). “Event-based MILP models for resource-constrained
575 project scheduling problems”. In: *Computers & Operations Research* 38.1, pp. 3–13.
- 576 Mingozzi, A., Maniezzo, V., Ricciardelli, S., and Bianco, L. (May 1998). “An Exact Algorithm for the Resource-
577 Constrained Project Scheduling Problem Based on a New Mathematical Formulation”. In: *Management Science*
578 44.5, pp. 714–729.

579 Morin, P.-A., Artigues, C., and Haït, A. (2017). “Periodically Aggregated Resource Constrained Project Scheduling
580 Problem”. In: *European Journal of Industrial Engineering* 11.6, pp. 792–817.

581 Okubo, H., Miyamoto, T., Yoshida, S., Mori, K., Kitamura, S., and Izui, Y. (May 2015). “Project Scheduling un-
582 der Partially Renewable Resources and Resource Consumption during Setup Operations”. In: *Computers &
583 Industrial Engineering* 83, pp. 91–99.

584 Paul, M. and Knust, S. (2015). “A Classification Scheme for Integrated Staff Rostering and Scheduling Problems”.
585 In: *RAIRO-Operations Research* 49.2, pp. 393–412.

586 Riedler, M., Jatschka, T., Maschler, J., and Raidl, G. R. (2020). “An iterative time-bucket refinement algorithm for
587 a high-resolution resource-constrained project scheduling problem”. In: *International Transactions in Opera-
588 tional Research* 27.1, pp. 573–613.

589 Schutt, A., Feydy, T., and Stuckey, P. J. (2013). “Explaining time-table-edge-finding propagation for the cumu-
590 lative resource constraint”. In: *International Conference on Integration of Constraint Programming, Artificial
591 Intelligence, and Operations Research*. Springer, pp. 234–250.

592 Appendix

593 A Proof of Theorem 4: formal correctness of formulation (F1)

594 *Proof.* We first show that a feasible solution for formulation (F1) is a feasible solution for the PARCPSP with the
595 same objective function value. Consider a solution S_i the MILP and suppose it is unfeasible for the PARCPSP. Since
596 constraints (8) translate directly the precedence constraints, the solution must be resource-unfeasible, which means
597 that constraints (3) of the conceptual model is violated. This can only be the case if there exists $i \in \mathcal{A}$ and $\ell \in \mathcal{L}$
598 such $d_{i,\ell} < \max\left(0, \min(S_i + p_i, \ell\Delta) - \max(S_i, (\ell - 1)\Delta)\right)$, i.e. $d_{i,\ell}$ is strictly smaller than the intersection length
599 of intervals $[S_i, S_i + p_i]$ and $[(\ell - 1)\Delta, \ell\Delta]$. Otherwise constraints (9) ensure that constraints (3) are satisfied. The
600 lower bound to $d_{i,\ell}$ for each $i \in \mathcal{A}$ and $\ell \in \mathcal{L}$ is set by its non-negativity and by constraints (12–14). The latter
601 constraints involve binary variables $z_{S_i, \ell-1}$ and $z_{f_i, \ell}$. We show that $d_{i,\ell}$ is not smaller than the intersection length for
602 each of the possible values for pair $(z_{S_i, \ell-1}, z_{f_i, \ell})$.

- 603 • $z_{S_i, \ell-1} = 0$ and $z_{f_i, \ell} = 0$. According to (10), since $z_{S_i, \ell-1} = 0$ we have $S_i \geq (\ell - 1)\Delta$. Similarly, with $z_{f_i, \ell} = 0$
604 constraint (11) yields $S_i + p_i \geq \ell\Delta$. In this case the intersection length is 0 if $S_i \geq \ell\Delta$ and $\ell\Delta - S_i$ otherwise.
605 This is ensured by the non-negativity of $d_{i,\ell}$ in conjunction with constraints (13).
- 606 • $z_{S_i, \ell-1} = 0$ and $z_{f_i, \ell} = 1$. As for the previous case, inserting $z_{S_i, \ell-1} = 0$ in (10) gives $S_i \geq (\ell - 1)\Delta$. Setting
607 $z_{f_i, \ell} = 1$ in (11) yields $S_i + p_i \leq \ell\Delta$. This is the case where interval $[S_i, S_i + p_i]$ is included in $[(\ell - 1)\Delta, \ell\Delta]$,
608 so the intersection length is equal to p_i . Remark that $S_i \geq (\ell - 1)\Delta \implies S_i + p_i > (\ell - 2)\Delta$. Hence, we
609 have $z_{f_i, \ell-2} = 0$ since $z_{f_i, \ell-2} = 1$ and (11) would imply that $S_i + p_i \leq (\ell - 2)\Delta$. Then, since $z_{f_i, \ell-2} = 0$
610 and $z_{S_i, \ell-1} = 0$, (12) yields $d_{i, \ell-1} = 0$. We have also $d_{i, \ell'} = 0$ for all $\ell' \leq \ell - 1$ because $z_{f_i, \ell'-1} = 0$ and
611 $z_{S_i, \ell'} = 0$ according to step constraints. Another remark is that $S_i + p_i \leq \ell\Delta \implies S_i < (\ell + 1)\Delta$. With (10),
612 this yields $z_{S_i, \ell+1} = 1$. Since in addition $z_{f_i, \ell} = 1$, constraint (12) yields $d_{i, \ell+1} = 0$. With step constraints,

613 we have $z_{f_i, \ell'-1} = 1$ and $z_{s_i, \ell'} = 1$ and so with (12) $d_{i, \ell'} = 0$ for all $\ell' \geq \ell + 1$. It follows that $d_{i, \ell'} = 0$ for all
614 $\ell' \in \mathcal{L} \setminus \{\ell\}$. According to constraint (15), we obtain $d_{i, \ell} = p_i$.

615 • $z_{s_i, \ell-1} = 1$ and $z_{f_i, \ell} = 0$. (10) and $z_{s_i, \ell-1} = 1$ implies that $S_i \leq (\ell - 1)\Delta$, while (11) and $z_{f_i, \ell} = 0$ imply that
616 $S_i + p_i \geq (\ell)\Delta$. In this case interval $[(\ell - 1)\Delta, \ell\Delta]$ is included in interval $[S_i, S_i + p_i]$ and the intersection length
617 is equal to Δ . Inserting $z_{s_i, \ell-1} = 1$ and $z_{f_i, \ell} = 0$ in (12) directly gives $d_{i, \ell} \geq \Delta$.

618 • $z_{s_i, \ell-1} = 1$ and $z_{f_i, \ell} = 1$. In this case, we obtain $S_i \leq (\ell - 1)\Delta$ with (10) and $S_i + p_i \leq \ell\Delta$ with (11). It follows
619 that the intersection length is $S_i + p_i - (\ell - 1)\Delta$ if $S_i + p_i \geq (\ell - 1)\Delta$ and 0 otherwise. Constraints (14) yields
620 $d_{i, \ell} \geq S_i + p_i - (\ell - 1)\Delta$.

621 As in all case $d_{i, \ell}$ is not smaller than the actual length of the intersection of intervals $[S_i, S_i + p_i]$ and $[(\ell - 1)\Delta, \ell\Delta]$,
622 any feasible solution of the MILP is also feasible for the PARCPSP. Furthermore the objective functions are exactly
623 the same.

624 It remains to show that for any feasible solution of the PARCPSP, there is a compatible assignment of the
625 other decision variables that satisfies all the constraints of the MILP. Let $S_i, i \in \mathcal{A}$ denote a feasible solution of
626 the PARCPSP. Obviously the precedence constraints (8) are satisfied. Setting variables $d_{i, \ell}$ to $d_{i, \ell}(S)$ according to
627 its definition in §2.2 allows to satisfy constraints (9) and (15). Consider the following assignment for variables
628 $z_{s_i, \ell}$. Let $l s^i$ the period such that $(l s^i - 1)\Delta \leq S_i < l s^i \Delta$. For all $i \in \mathcal{A}$, let us set $z_{s_i, \ell} = 0$ for each $\ell < l s^i$ and
629 $z_{s_i, \ell} = 1$ for each $\ell \geq l s^i$. Similarly, let $l f^i$ the period verifying $(l f^i - 1)\Delta \leq S_i + p_i < l f^i \Delta$. For all $i \in \mathcal{A}$, let us
630 set $z_{f_i, \ell} = 0$ for each $\ell < l f^i$ and $z_{f_i, \ell} = 1$ for each $\ell \geq l f^i$. This assignment obviously satisfies the step behavior
631 constraints of variables $z_{s_i, \ell}$ and $z_{f_i, \ell}$. Start time lower bound constraints (10) are satisfied as they give $S_i \geq \ell\Delta$ for
632 $\ell = 1, \dots, l s^i - 1$ and $S_i \geq 0$ for $\ell = l s^i, \dots, L$. Start time upper bound constraints (10) are also satisfied as they can
633 be written $S_i \leq L\Delta$ for $\ell = 1, \dots, l s^i - 1$ and $S_i \leq \ell\Delta$ for $\ell = l s^i, \dots, L$. The same holds for completion time lower
634 and upper bound constraints (11), since we obtain $\ell\Delta \leq S_i + p_i \leq L\Delta$ for $\ell = 1, \dots, l f^i - 1$ and $0 \leq S_i + p_i \leq \ell\Delta$
635 for $\ell = l f^i, \dots, L$. Now, let us consider the constraints (10-14) that link $d_{i, \ell}, S_i, z_{s_i, \ell}$ and $z_{f_i, \ell}$ variables. Recall
636 that $d_{i, \ell}$ is set to $\max(0, \min(S_i + p_i, \ell\Delta) - \max(S_i, (\ell - 1)\Delta))$. For each task i , we consider the following sets
637 $L1 = \{\ell \in \mathcal{L} | \ell < l s^i\}$, $L2 = \{\ell \in \mathcal{L} | \ell > l f^i\}$ and $L3 = \{l s^i + 1, \dots, l f^i - 1\}$. Note that $\mathcal{L} = L1 \cup \{l s^i, l f^i\} \cup L2 \cup L3$.
638 By definition of $l s^i$ and $l f^i$, $d_{i, l s^i} = l s^i \Delta - S_i$ and $d_{i, l f^i} = S_i + p_i - (l f^i - 1)\Delta$ if $l f^i > l s^i$ and $d_{i, l f^i} = d_{i, l s^i} = p_i$
639 otherwise. For $\ell \in L1 \cup L2$, $d_{i, \ell} = 0$. For $\ell \in L3$, $d_{i, \ell} = \Delta$. We show below that constraints (13-14) are all compatible
640 with these values.

641 Non-negativity constraints are satisfied for all $\ell \in \mathcal{L}$. Constraints (12) are equivalent to $d_{i, \ell} \leq 0$ for $\ell \in L1 \cup L2$
642 and $d_{i, \ell} \leq \Delta$ for $\ell = \{l s^i, l f^i\} \cup L3$. Constraints (12) can be written $d_{i, \ell} \geq 0$ for $\ell \in L1 \cup L2 \cup \{l s^i, l f^i\}$ and $d_{i, \ell} \geq \Delta$ for
643 $\ell \in L3$. Constraints (13) give $d_{i, \ell} \geq \ell\Delta - S_i < 0$ for $\ell \in L1$, $d_{i, \ell} \geq l s^i \Delta - S_i$ for $\ell = l s^i$ in the case where $z_{f_i, l s^i} = 0$ (i.e.
644 $l f^i > l s^i$) and $d_{i, \ell} \geq (l s^i - 1)\Delta - S_i < 0$ for $\ell = l s^i$ in the case where $z_{f_i, l s^i} = 1$. For $\ell \in L3$, we obtain $d_{i, \ell} \geq -S_i$. For
645 $\ell = l f^i > l s^i$ and for $\ell \in L2$, we have $d_{i, \ell} \geq -\Delta - S_i$. Last, constraints (14) give precisely $d_{i, \ell} \geq S_i + p_i - (l f^i - 1)\Delta$
646 for $\ell = l f^i$ and $l f^i > l s^i$. For $\ell = l f^i = l s^i$, we have $d_{i, \ell} \geq S_i + p_i - l f^i \Delta < 0$. For $\ell \in L1$ or $\ell = l s^i < l f^i$, the
647 constraints is written $d_{i, \ell} \geq S_i + p_i - (L + 1)\Delta < 0$. For $\ell \in L3$, we obtain $d_{i, \ell} \geq S_i + p_i - L\Delta < 0$. For $\ell \in L2$, the
648 constraint yields $d_{i, \ell} \geq S_i + p_i - (\ell - 1)\Delta < 0$.

649 □

650 B Proof of Theorem 5: formal correctness of formulation (F2)

651 *Proof.* Given the common structure with the first formulation and the definition of variables $d_{i,\ell}$, $\lambda_{i,\ell}$ and $\mu_{i,\ell}$, we
652 just have to show that constraints (27–36) properly model the relationships $d_{i,\ell} = [S_i, S_i + p_i] \cap [(\ell - 1)\Delta, \ell\Delta]$ for
653 each activity i . Constraints (30–31) can be rewritten $z_{i,\ell+1}^\lambda \leq \frac{\lambda_{i,\ell}}{\Delta} \leq z_{i,\ell}^\lambda$, meaning that variables $z_{i,\ell}^\lambda$ have a decreasing
654 step behavior. Suppose that a period ℓ is such that $\ell\Delta \geq S_i$ and $\lambda_{i,\ell} > 0$. Then we have $z_{i,\ell}^\lambda = 1$ and so $z_{i,\ell'}^\lambda = 1$ and
655 $\lambda_{i,\ell'} = \Delta$, for all $\ell' < \ell$. In this case we would have $\sum_{\ell'=1}^{\ell} \lambda_{i,\ell'} > S_i$, a contradiction. It follows that any non zero
656 $\lambda_{i,\ell}$ variable is such that $\ell\Delta < S_i$. It follows that if $S_i > 0$, the first $\ell \in \{1, \dots, \lfloor \frac{S_i}{\Delta} \rfloor\}$ periods are such that $\lambda_{i,\ell} = \Delta$
657 and period $\ell = \lfloor \frac{S_i}{\Delta} \rfloor + 1$ is such that $\lambda_{i,\ell} = S_i \bmod \Delta$. We have precisely $\lambda_{i,\ell} = [0, S_i] \cap [(\ell - 1)\Delta, \ell\Delta]$ for all $\ell \in \mathcal{L}$.
658 Similarly, constraints (32–33) yield $z_{i,\ell-1}^\mu \leq \frac{\mu_{i,\ell}}{\Delta} \leq z_{i,\ell}^\mu$ for all $\ell \in \mathcal{L}$. Hence variables $z_{i,\ell-1}^\mu$ have an increasing step
659 behavior. Composition of constraints (27), (28) and (29) give $L\Delta - S_i - p_i = \sum_{\ell \in \mathcal{L}} \mu_{i,\ell}$. With the same reasoning
660 it comes that $\mu_{i,\ell} = [S_i + p_i, L\Delta] \cap [(\ell - 1)\Delta, \ell\Delta]$. From constraints (27), we obtain $d_{i,\ell} = \Delta - [S_i + p_i, L\Delta] \cap [(\ell -$
661 $1)\Delta, \ell\Delta] - [0, S_i] \cap [(\ell - 1)\Delta, \ell\Delta] = [S_i, S_i + p_i] \cap [(\ell - 1)\Delta, \ell\Delta]$. \square

662 C Proof details of Theorem 7

663 *Proof.* We provide below the full proof of the implication of model (F1s) by model (F2s) for Constraints (10)
664 and (11) (link between S_i , $z_{S_i,\ell}$ and $z_{f_i,\ell}$) and Constraints (12) to (14) (expression of $d_{i,\ell}$). For each constraint, the
665 variables of model (F1s) are substituted by the variable of model (F2s), which yields the constraint with a prime (')
666 that are then shown to be always satisfied.

$$S_i \leq (\ell - 1)\Delta + (L - \ell + 1)\Delta z_{i,\ell}^\lambda \quad \text{using } 10_{UB} \text{ for } l - 1 \text{ and } z_{S_i,\ell-1} = 1 - z_{i,\ell}^\lambda \quad (10'_{UB})$$

$$\begin{aligned} & S_i - (\ell - 1)\Delta - (L - \ell + 1)\Delta z_{i,\ell}^\lambda \\ &= -(\ell - 1)\Delta + \left(\sum_{\ell'=1}^L \lambda_{i,\ell'}\right) - \left(\sum_{\ell'=\ell}^L \Delta\right) z_{i,\ell}^\lambda \\ &\leq -\sum_{\ell'=\ell}^L \left(\Delta z_{i,\ell'}^\lambda - \lambda_{i,\ell'}\right) - \sum_{\ell'=1}^{\ell-1} (\Delta - \lambda_{i,\ell'}) \\ &\leq 0 \end{aligned}$$

$$S_i + p_i \geq \ell\Delta - \ell\Delta z_{i,\ell}^\mu \quad (11'_{LB})$$

$$\begin{aligned} & S_i + p_i - \ell\Delta + \ell\Delta z_{i,\ell}^\mu \\ &= -\ell\Delta + (L\Delta - \sum_{\ell'=1}^L \mu_{i,\ell'}) + \left(\sum_{\ell'=1}^{\ell} \Delta\right) z_{i,\ell}^\mu \\ &\geq \sum_{\ell'=1}^{\ell} \left(\Delta z_{i,\ell'}^\mu - \mu_{i,\ell'}\right) + \sum_{\ell'=\ell+1}^L (\Delta - \mu_{i,\ell'}) \\ &\geq 0 \end{aligned}$$

$$S_i + p_i \leq \ell\Delta + (L - \ell)\Delta \left(1 - z_{i,\ell}^\mu\right) \quad (11'_{UB})$$

$$\begin{aligned}
& S_i + p_i - \ell\Delta - (L - \ell)\Delta \left(1 - z_{i,\ell}^\mu\right) \\
&= -\ell\Delta + (L\Delta - \sum_{\ell'=1}^L \mu_{i,\ell'}) - (L - \ell)\Delta + (\sum_{\ell'=\ell+1}^L \Delta) z_{i,\ell}^\mu \\
&\leq -\sum_{\ell'=1}^L \mu_{i,\ell'} + \sum_{\ell'=\ell+1}^L \Delta z_{i,\ell'-1}^\mu \\
&\leq -\sum_{\ell'=1}^L \mu_{i,\ell'} + \sum_{\ell'=\ell+1}^L \mu_{i,\ell'} \\
&= -\sum_{\ell'=1}^{\ell} \mu_{i,\ell'} \\
&\leq 0
\end{aligned}$$

$$d_{i,\ell} \geq \Delta \left(1 - z_{i,\ell}^\lambda - z_{i,\ell}^\mu\right) \quad (12'_{LB})$$

$$\begin{aligned}
& d_{i,\ell} - \Delta \left(1 - z_{i,\ell}^\lambda - z_{i,\ell}^\mu\right) \\
&= \Delta - \lambda_{i,\ell} - \mu_{i,\ell} - \Delta + \Delta z_{i,\ell}^\lambda + \Delta z_{i,\ell}^\mu \\
&= \left(\Delta z_{i,\ell}^\lambda - \lambda_{i,\ell}\right) + \left(\Delta z_{i,\ell}^\mu - \mu_{i,\ell}\right) \\
&\geq 0
\end{aligned}$$

$$d_{i,\ell} \leq \Delta \left(1 - z_{i,\ell+1}^\lambda - z_{i,\ell-1}^\mu\right) \quad (12'_{UB})$$

$$\begin{aligned}
& d_{i,\ell} - \Delta \left(1 - z_{i,\ell+1}^\lambda - z_{i,\ell-1}^\mu\right) \\
&= \Delta - \lambda_{i,\ell} - \mu_{i,\ell} - \Delta + \Delta z_{i,\ell+1}^\lambda + \Delta z_{i,\ell-1}^\mu \\
&= \left(\Delta z_{i,\ell+1}^\lambda - \lambda_{i,\ell}\right) + \left(\Delta z_{i,\ell-1}^\mu - \mu_{i,\ell}\right) \\
&\leq 0
\end{aligned}$$

$$d_{i,\ell} \geq \ell\Delta - S_i - \Delta z_{i,\ell}^\mu - \ell\Delta \left(1 - z_{i,\ell}^\lambda\right) \quad (13')$$

$$\begin{aligned}
& d_{i,\ell} - \ell\Delta + S_i + \Delta z_{i,\ell}^\mu + \ell\Delta \left(1 - z_{i,\ell}^\lambda\right) \\
&= d_{i,\ell} - \ell\Delta + (\sum_{\ell'=1}^L \lambda_{i,\ell'}) + \Delta z_{i,\ell}^\mu + \ell\Delta - \ell\Delta z_{i,\ell}^\lambda \\
&= \sum_{\ell'=1}^L \lambda_{i,\ell'} + d_{i,\ell} + \Delta \left(z_{i,\ell}^\mu - z_{i,\ell}^\lambda\right) - (\sum_{\ell'=1}^{\ell-1} \Delta) z_{i,\ell}^\lambda \\
&\geq \sum_{\ell'=1}^L \lambda_{i,\ell'} + (\Delta - \lambda_{i,\ell} - \mu_{i,\ell}) + \Delta \left(z_{i,\ell}^\mu - z_{i,\ell}^\lambda\right) - \left(\sum_{\ell'=1}^{\ell-1} \Delta z_{i,\ell'+1}^\lambda\right) \\
&\geq (\sum_{\ell'=1}^L \lambda_{i,\ell'} - \lambda_{i,\ell} - \sum_{\ell'=1}^{\ell-1} \lambda_{i,\ell'}) + \left(\Delta z_{i,\ell}^\mu - \mu_{i,\ell}\right) + \Delta \left(1 - z_{i,\ell}^\lambda\right) \\
&\geq \sum_{\ell'=\ell+1}^L \lambda_{i,\ell'} \\
&\geq 0
\end{aligned}$$

$$d_{i,\ell} \geq S_i + p_i - (\ell - 1)\Delta - \Delta z_{i,\ell}^\lambda - (L - \ell + 1)\Delta \left(1 - z_{i,\ell}^\mu\right) \quad (14')$$

$$\begin{aligned}
& d_{i,\ell} - S_i - p_i + (\ell - 1)\Delta + \Delta z_{i,\ell}^\lambda + (L - \ell + 1)\Delta \left(1 - z_{i,\ell}^\mu\right) \\
&= d_{i,\ell} + (\ell - 1)\Delta - (L\Delta - \sum_{\ell'=1}^L \mu_{i,\ell'}) + \Delta z_{i,\ell}^\lambda + (L - \ell + 1)\Delta - (L - \ell + 1)\Delta z_{i,\ell}^\mu \\
&= \sum_{\ell'=1}^L \mu_{i,\ell'} + d_{i,\ell} + \Delta \left(z_{i,\ell}^\lambda - z_{i,\ell}^\mu\right) - \left(\sum_{\ell'=\ell+1}^L \Delta\right) z_{i,\ell}^\mu \\
&\geq \sum_{\ell'=1}^L \mu_{i,\ell'} + (\Delta - \lambda_{i,\ell} - \mu_{i,\ell}) + \Delta \left(z_{i,\ell}^\lambda - z_{i,\ell}^\mu\right) - \left(\sum_{\ell'=\ell+1}^L \Delta z_{i,\ell'-1}^\mu\right) \\
&\geq \left(\sum_{\ell'=1}^L \mu_{i,\ell'} - \mu_{i,\ell} - \sum_{\ell'=\ell+1}^L \mu_{i,\ell'}\right) + \left(\Delta z_{i,\ell}^\lambda - \lambda_{i,\ell}\right) + \Delta \left(1 - z_{i,\ell}^\mu\right) \\
&\geq \sum_{\ell'=1}^{\ell-1} \mu_{i,\ell'} \\
&\geq 0
\end{aligned}$$