New MINLP Formulations for the Unit Commitment Problems with Ramping Constraints

Tiziano Bacci
Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", C.N.R. Via dei Taurini 19, 00185 Rome – Italy
tiziano.bacci@iasi.cnr.it

Antonio Frangioni
Dipartimento di Informatica, Università di Pisa Largo B. Pontecorvo 3, 56127 Pisa – Italy frangio@di.unipi.it

Claudio Gentile
Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", C.N.R. Via dei Taurini 19, 00185 Rome – Italy
gentile@iasi.cnr.it

Kostas Tavlaridis-Gyparakis
Dipartimento di Informatica, Università di Pisa Largo B. Pontecorvo 3, 56127 Pisa – Italy kostas.tavlaridis@gmail.com

The Unit Commitment (UC) problem in electrical power production requires to optimally operate a set of power generation units over a short time horizon. Operational constraints of each unit depend on its type and can be rather complex. For thermal units, typical ones concern minimum and maximum power output, minimum up- and down-time, start-up and shut-down limits, ramp-up and ramp-down limits, non-linear objective function. In this work we present the first MINLP formulation that describes the convex hull of the feasible solutions of the *single-unit commitment problem* (1UC) comprising all the above constraints, and convex power generation costs. The new formulation has a polynomial number of both variables and constraints, and it is based on the efficient Dynamic Programming algorithm proposed in **?** together with the *perspective reformulation* technique proposed in **?**. The proof that the formulation is exact is based on a new extension of a result previously only available in the polyhedral case which is potentially of interest in itself. We then analyze the effect of using it to develop tight formulations for the more general (UC). Since the formulation is rather large, we also propose two new formulations, based on partial aggregations of variables, with different trade-offs between quality of the bound and cost of the solving the continuous relaxation. Our results show that navigating these trade-offs may lead to improved performances.

Key words: Unit Commitment problem, Ramp Constraints, MIP Formulations, Dynamic Programming, Convex Costs

1. Introduction

The Unit Commitment (UC) problem is a basic problem arising in power industries to coordinate and manage power generation units. Although it was the typical problem to be solved in old monopolistic regimes, the need to solve UC problems has not disappeared. On the contrary, both generation companies and system operators need to routinely solve some UC variant even in the free market regime (e.g., see **?????**), both before and after the price of energy (and other ancillary services) has been cleared in the relevant market. Due to the huge figures involved in real-world systems (**?**), even minor improvements on the quality of the obtained solutions can result in very significant economical (and, possibly, environmental) savings. Therefore, the efficient solution of UC problems is still very much relevant in practice. Besides, UC being a complex Mixed-Integer NonLinear Program (MINLP), its study is relevant from the methodological viewpoints. Indeed, some theoretical results (e.g., **?**) that have been originally motivated by UC have later found many more applications (**??????**).

The traditional UC problem requires finding the schedule of each power generation unit in order to minimize operational costs while satisfying both system-wide constraints and operational constraints associated with each unit. System-wide constraints usually comprise the satisfaction of the energy demand, the provision of different types of reserve, and the handling of the transmission network. Operational constraints depend on the type of generation units. Most power systems are mainly based on thermal units (comprised nuclear ones) and hydro units, but in recent years the contribution of renewable energy sources (wind, solar, etc.) has steadily increased. As these are characterized by uncertainty in the production output, uncertain (robust and/or stochastic) UC models are more and more necessary (**??**). Since uncertain variants of optimization problems are typically considerably more difficult to solve than deterministic ones, efficient solution methods for these problems are in high demand as much as ever. Moreover, there is a clear trend whereby production and consumption tend to become more geographically separated than they previously were (think offshore wind farms and large solar plants in semi-desert areas), which is putting novel strain on transmission networks that were not originally planned for these scenarios. All this has a substantial impact both on costs and security/reliability (**??**), providing further strong motivation for the development of new approaches capable of solving complex variants of UC problems in shorter and shorter computational times.

Traditionally, Lagrangian relaxation was the method of choice to solve UC (e.g., see **???**, **?**, §3.3), since it was capable of exploiting the spatial structure of the problem: the most complex constraints pertain to the behaviour of a single unit, and relatively fewer and simpler ones link the different units together. However, the advances in the solution of Mixed-Integer (linear and convex) Programming (MIP) problems that are now widely available in present commercial solver have made MIP approaches an attractive option. This is even more so as the two approaches can be fruitfully combined (**??**). The first MILP formulation for UC was described in **?** and used three sets of binary variables. Later on, formulations using only one set of binary variables (on/off state) became more popular (**??**), although different ones continued to be used (**?**). While a reduced number of variables may lessen the cost of computing relaxations, this is usually not the crucial factor; rather, the *tightness* (quality of the lower bound of the continuous relaxation) of a MIP formulation is key for the efficient solution of the problem.

As operational constraints of thermal units have a strong combinatorial structure, many efforts have been made to improve their MIP definition. While there are some different types of units, each with several different variants, thermal units (comprised nuclear ones) are still the bulk of most energy systems, and these most often have a common set of operational restrictions. In particular, constraints on minimum and maximum power output, minimum up- and down-time, and ramp-up/ramp-down in power are almost invariably imposed. Most units have a nonzero minimum power production (and, obviously, a maximum one), meaning that the produced power is a typical *semi-continuous* variable, which can either be 0 or live in a closed real interval. These are naturally modeled with the help of the (also natural) binary variables that dictate if the unit is on or off. Minimum up- and down-time constraints establish a minimum number of consecutive time periods that a unit must be on, or off; they are typically imposed to limit technical stress of the thermal units due to frequent start-up and shut-down operations. Such constraints introduce a strong combinatorial structure. The first exact description by means of linear inequalities for minimum-up and -down time constraints has been given in **?** with an exponential number of inequalities and a polynomial time separation algorithm. Afterwards, **?** and independently **?** developed an extended linear description with a linear number of constraints. Ramp-up and ramp-down constraints limit the maximum increase or decrease of the power production between two consecutive time periods. Moreover, in these same linear inequalities also maximum limits on start-up and shut-down periods are often imposed. Further complex features of thermal units are related to power production costs. The cost of producing energy is typically a nonlinear function of the produced energy already when the unit is in the stable state ("on"). Furthermore, usually start-up costs have to be paid in the period when the unit is started up, to account for the nontrivial start-up operations. In their simplest description start-up costs can be considered fixed, but in a more exact description they are dependent on how long the unit remained off before start-up. This is because, roughly speaking, the unit must reach a minimum temperature in order to be able to produce power, and the heating process requires energy that has to be paid for. The cost for reaching the required temperature depends on the temperature that the unit has when the start-up process begins, which in turn depends both on environmental factors (assumed known) and on the previous history of the unit, i.e., how many periods it has been off.

In this work we present the first MIP description of the convex hull of the solutions satisfying *all* the standard operational constraints for the thermal units: minimum up- and down-time constraints, minimum and maximum power output, ramp constraints (including start-up and shut-down limits), general start-up costs, and nonlinear convex power production costs. Our new formulation is derived by a Dynamic Programming algorithm **?** and contains a polynomial number of variables and constraints. A simpler version of this result, limited to linear power generation costs, was first presented in **?**. Analogous results, using different proof techniques and still limited to (piecewise-)linear production costs, where independently proposed in **??**. Interestingly, the latter paper claims that the proposed formulation represents the convex hull for any possible convex power generation cost, but this is proven to be false (even for simple quadratic separable cost) in **??**. Indeed, nonlinearity of the cost function introduces a further complexity in the convex hull description, that none of the previous attempts addressed. This complexity can be tamed with the help of the *perspective reformulation* technique, that was originally developed in the seminal paper **?** precisely in the context of the solution of UC problems. Yet, the technique has rapidly found a surprisingly large number of other applications where it has been shown to significantly improve the computational performances of exact and heuristic approaches to MINLP with semicontinuous variables and/or disjunctive constraints **????????**. Perhaps more relevantly for this work, perspective reformulation techniques lie at the heart of several results, among which recent ones such as **?**, that allow to describe the convex hull of mixed-integer nonlinear regions with different properties. We contribute to this growing body of knowledge by extending the well-known "Approach no. 4" of **?** (used by **?** and by others) to the nonlinear case. This allows to prove that a formulation obtained by "mixing" in a proper way individual sub-models, each of which properly represents the convex hull of their integer partial solution, yields an exact overall formulation. This is a nontrivial result potentially of interest in other contexts; for instance, the results in **?** provide convex hull descriptions under the condition (among others) that the constraints only involve the binary variables, and therefore could be used to construct the individual "pieces" of an overall exact formulation that can then be constructed with the technique developed here.

The structure of the paper is as follows. In Section **??** we recall the main formulation of the UC problem. In Section **??** we give a survey of the main results concerning polyhedral results for UC formulations. In Section **??** we present an improved version of the Dynamic Programming algorithm described in **?**. In Section **??** we present the new formulation and we prove that it describes the convex hull of the solutions of the single-unit commitment problem (1UC). To this aim we define the 1-sum composition of MINLPs and prove that it preserves exactness of formulations, a not unexpected but still new and nontrivial result. In Section **??** we propose two additional simplified formulations also based on the Dynamic Programming algorithm with a trade-off between tightness and compactness. In Section **??** we present some preliminary computational experiments aimed at gauging the practical effectiveness of the new formulations on a data set already used in the literature. Finally, in Section **??** we sum up the results, and draw some possible lines for future research on the topic.

2. The Thermal Unit Commitment Problem

In this section we recall the most popular MIP formulation of the thermal Unit Commitment problem (other kinds of units have entirely different constraints and therefore require specific study, e.g. **?**). This formulation is usually named *3-bin* formulation from the number of vectors of binary variables that are considered and to distinguish it from the *1-bin* formulation that contains only one vector of binary variables.

Let I be the set of (indices of) thermal generators, with $m = |I|$, and $T = \{1, \dots, n\}$ be the set of (indices of) time periods in the planning horizon. Given two time periods t' and t'', we will denote by $T(t', t'')$ the set of all the time periods between t' and t'', extremes included (obviously, $T(t', t'') = \emptyset$ if $t' > t''$). For each $i \in I$ and $t \in T$, let p_{it} (the *power variables*) be the power level of unit i at time period t, and x_{it} (the *commitment variables*) be the binary variable denoting the on/off state of unit i at time period t. As previously recalled, "on" state means that the unit can be *modulated*, i.e., the power output of the unit can be increased or decreased subject to some technical constraints. The "off" state does not necessarily mean that the unit is inactive, in that it could be performing a start-up or shut-down trajectory, or being "banking", i.e., burning fuel to keep the temperature of the unit in view of an imminent restart. All these details are largely transparent to our model: although complex start-up or shut-down trajectories may require some modifications to be completely accounted for (**?**), this does not impact the mathematical formulation of the individual units. The *3-bin* formulation has been introduced in **?**, and independently in **?**. It starts from an exact formulation of the minimum up- and down-time constraints only, that is obtained by considering 3 vectors of binary variables: besides the commitment variables x_{it}, one introduces *start-up variables* v_{it} denoting if unit i has been started up at time period t (i.e., $x_{it} = 1$ and $x_{i,t-1} = 0$) and *shut-down variables* w_{it} denoting if i has been shut-down t (i.e., $x_{it} = 0$ and $x_{i,t-1} = 1$). Let τ_i^+ and τ_i^- being the minimum number of consecutive time periods that unit i has to be in on and off state, respectively, using the above defined variables, the minimum up- and down-time constraints can be defined by

$$\textstyle\sum_{s \in T(t-\tau_+^i+1,t)} v_{is} \leq x_{it} \qquad t \in T(\tau_i^+, n), \tag{1}$$

$$\textstyle\sum_{s \in T(t-\tau_-^i+1,t)} w_{is} \leq 1 - x_{it} \qquad t \in T(\tau_i^-, n), \tag{2}$$

$$x_{it} - x_{i,t-1} = v_{it} - w_{it} \qquad t \in T(1, n); \tag{3}$$

(note how (**??**) are flow-conservation-type constraints). The first exact description of these conditions were given in **?** using only the commitment variables (*1-bin* formulation), but with an exponential number of inequalities and a polynomial time separation algorithm.

Further constraints are required to specify the initial conditions of the unit. Let τ_i^0 denote the initial state of unit i as follows: at the beginning of the planning horizon, if $\tau_i^0 > 0$ then unit i has been in on state for τ_i^0 time periods, thus one has to impose the condition:

$$x_{it} = 1 \qquad t \in T(1, \tau_i^+ - \tau_i^0)\ . \tag{4}$$

Of course, this is only required if, besides $\tau_i^0 > 0$, one also has $\tau_i^0 < \tau_i^+$ (otherwise, $T(1, \tau_i^+ - \tau_i^0) = \emptyset$). Similarly, $\tau_i^0 < 0$ means that unit i has been in off state for $-\tau_i^0$ time periods, and one has to impose the condition (again, this is only significant if $-\tau_i^0 < \tau_i^-$):

$$x_{it} = 0 \qquad t \in T(1, \tau_i^- + \tau_i^0)\ . \tag{5}$$

With l_i and u_i being the minimum and the maximum power output for unit $i \in I$, respectively, the minimum and maximum power output constraints are simply

$$l_i x_{it} \leq p_{it} \leq u_i x_{it} \qquad t \in T. \tag{6}$$

Let now Δ_i^+ and Δ_i^- be the ramp-up and ramp-down limits for unit i, respectively, i.e., the maximum increase/decrease of power output w.r.t. the previous period. The corresponding ramp constraints would be quite simple to write, were it not for the special treatment required by the start-up and shut-down periods. It is expedient to introduce two specific values $\bar{l}_i$ and $\bar{u}_i$, known as the start-up and shut-down limits for unit i. These are the maximum power value that the unit can have in a, respectively, start-up and shut-down period, and they can be different from l_i and u_i; for consistency, it must however be $l_i \leq \bar{l}_i \leq u_i$ and $l_i \leq \bar{u}_i \leq u_i$. Then, the ramp constraints can be formulated as follows (**?**):

$$p_{it} - p_{i,t-1} \leq \Delta_i^+ x_{i,t-1} + \bar{l}_i v_{it} \qquad t \in T, \tag{7}$$

$$p_{i,t-1} - p_{it} \leq \Delta_i^- x_{it} + \bar{u}_i w_{it} \qquad t \in T. \tag{8}$$

We can also assume that $0 \leq \Delta_i^+ \leq u_i - l_i$ and $0 \leq \Delta_i^- \leq u_i - l_i$, otherwise constraints (**??**)–(**??**) are either redundant or not feasible. Note that for $t = 1$ the constraints (**??**)–(**??**) refer to values p_{i0} and x_{i0}, which clearly are not variables but parameters to be set according to the initial conditions (cf. τ_i^0 above). The objective function usually contains the minimization of the production costs, that depend on two main contributions: the generation costs and the start-up costs. The generation costs, for each unit i and time period t, are often expressed by a convex quadratic cost function of the type

$$f_i(p_{it}) = a_i p_{it}^2 + b_i p_{it} \ , \tag{9}$$

possibly plus a fixed cost $c_i x_{it}$. This is an approximation of the true cost function, that does not take into account some technical characteristics of the units, such as the so-called “valve points”. However, the approximation is generally deemed to be accurate enough for practical purposes. Indeed, in many cases the cost function is further approximated by a piecewise linear (or even downright linear) function in order to get good feasible solutions in short time (**?**).

Nevertheless (**??**) is convex, p_{it} is a semi-continuous variable generating a nonconvex feasible set. In other words, the "true" objective function should rather be expressed as

$$f_i(p_{it}, x_{it}) = \begin{cases} a_i p_{it}^2 + b_i p_{it} & \text{if } l_i \le p_{it} \le u_i \text{ and } x_{it} = 1 \\ 0 & \text{if } p_{it} = x_{it} = 0 \\ \infty & \text{otherwise} \end{cases} . \tag{10}$$

While (**??**) is nonconvex, its *convex envelope* (best possible convex approximation) turns out to be easily computed (**?**):

$$h_i(p_{it}, x_{it}) = \begin{cases} a_i \frac{p_{it}^2}{x_{it}} + b_i p_{it} & \text{if } x_{it} > 0 \text{ and } 0 < p_{it} \le u_i x_{it} \\ 0 & \text{if } p_{it} = x_{it} = 0 \\ \infty & \text{otherwise.} \end{cases} \tag{11}$$

This is called the *perspective function* of f_i; note that $h_i(p_{it}, x_{it}) = f_i(p_{it})$ if $x_{it} \in \{0, 1\}$, but $h_i(p_{it}, x_{it}) > f_i(p_{it})$ if $0 < x_{it} < 1$. Thus, substituting (**??**) to (**??**)—a technique known as *perspective reformulation*—has the potential to significantly increase the lower bound, as confirmed in several studies (**????????????**). Different special methods (**?????**) have been studied to efficiently deal with this "very nonlinear" term in the continuous relaxation without increasing too much its computational cost w.r.t. the case of the "simple" (**??**).

The start-up costs should in general be expressed as a function $s_i(x_i)$ of the complete state vector x_i, as it depends on the time τ that unit i has been off. In its most accurate formulation, the start-up cost can be computed by means of two functions. One is a concave cost function of the type $\sigma_i(\tau) = \overline{\sigma}_i(1 - e^{-\beta^i \tau}) + \alpha_i$, corresponding to the fact that the cost of starting up the unit depends on the temperature, which, if the unit is left to cool, drops with an exponential law towards ambient temperature (e.g., see **???**). However, for shorter stops it might be preferable to spend some fuel just in order to keep the unit at the right temperature, which can be assumed to have a linear cost $\gamma_i \tau$ on the number of time periods. For each value of τ, then, the optimal choice between the two options (usually referred to as "cooling" and "banking") is just the one giving minimum start-up cost. For our purposes, this complex function only need to be known at the discrete set of values

$$\sigma_{i\tau} = \min(\,\overline{\sigma}_i(1 - e^{-\beta_i \tau}) + \alpha_i \;,\; \gamma_i \tau\,) \qquad \tau \in T(\tau_i^-, \overline{\tau}_i) \;, \tag{12}$$

where $\overline{\tau}_i$ is the time such that $\overline{\sigma}_i(\overline{\tau}_i) \approx \overline{\sigma}_i(\overline{\tau}_i + 1) \approx \overline{\sigma}_i + \alpha_i$, i.e., the unit has reached ambient temperature and the start-up cost is maximum. Whatever the exact form of the function, the only relevant property needed for MIP formulations is that the values $\sigma_{i\tau}$ are non decreasing with respect to τ. Using this property, start-up costs can be expressed by means of a single extra new variable s_{it} and $\overline{\tau}_i - \tau_i^- + 1$ extra constraints (for each unit and time instant), as follows (**?**):

$$s_i(x_i) = \textstyle\sum_{t=1}^{n} s_{it} \tag{13}$$

$$s_{it} \geq \sigma_{i\tau}(v_{it} - \textstyle\sum_{j=2}^{\tau} w_{i,t-j+1}) \qquad t \in T \;,\; \tau \in T(\tau_i^-, \overline{\tau}_i) \;. \tag{14}$$

$$s_{it} \geq 0 \qquad t \in T. \tag{15}$$

Even though the number of extra variables and constraints in (**??**)–(**??**) is reasonably limited, using such a detailed representation of the start-up cost in a MIP model can have a substantial impact on the performances; this is why, most often the start-up costs are simply approximated with the fixed maximal cost ($\overline{\sigma}_i + \alpha_i$). In general, since solution time is a crucial issue, the trade-off between an accurate representation of the physical behavior of generating units and the solution cost of the corresponding models is nontrivial. In practice, often simplified models are employed in order to quickly find approximated solutions of good quality. We will refer to the parameters $\sigma_{i\tau}$ as *history-dependent start-up costs* if $\bar{\tau}_i > \tau_i^-$, while we will refer to *fixed start-up costs* when $\bar{\tau}_i = \tau_i^-$.

Note that with fixed start-up costs s_i, the *3-bin* formulation can be significantly simplified, as the start-up cost is then completely captured by adding the simple term

$$\textstyle\sum_{t \in T} s_i v_{it} \tag{16}$$

to the objective function, with no need of the extra variables s_{it} and the constraints (**??**).

While most of the constraints of the standard UC problem concern the behavior of a specific unit $i \in I$, system-wide constraints that link the decisions of the different units are also present. The simplest and most common form of system-wide constraints is that of the demand constraints

$$\sum_{i \in I} p_{it} = d_t \qquad t \in T, \tag{17}$$

where d_t is the (forecasted) total energy demand at time period t. These constraints are valid for the so called *bus network*, i.e., the case in which the transmission network has ample capacity to accommodate energy transfer and therefore the physical location of generators and constraints is irrelevant. In some (but not all) applications the capacity of the transmission network may become a limiting factor impacting the production decisions, and more accurate representations of the network are needed. The simplest ones (DC model) boil down to just a set of linear constraints, while the most accurate ones (AC model) involve highly nonlinear terms that are much harder to deal with. This has recently motivated a quite active research stream where formulations (or tight relaxations) of AC constraints are proposed using Second-Order Cone or SemiDefinite constraints (e.g., **???** and the references therein). Other system-wide constraints pertain to "reserve" (primary, secondary, or inertia) that are established to guarantee that the system will remain operational even if the actual conditions deviate (not too much) from the expected ones. Yet, all these variants typically do not impact of the formulation of the individual units, and therefore need not be discussed here in detail.

As the *3-bin* formulation is generally accepted as a better starting formulation, we adopt it as a benchmark for the new formulations that we will propose in the following. In particular, in the rest of the paper we will refer to the model T summarized in Table 3 in **?** as the *3-bin* formulation. The model T by **?** is based on the power variables p'_{it} defined according to the equation $p_{it} = p'_{it} + l_i x_{it}$ and it is actually the highest performing version of the *3-bin* formulation presented in the literature. We report in the Appendix the entire model T used in the experiments and denoted as *3-bin*.

3. Literature review of polyhedral descriptions

Here we revise the main polyhedral results proposed in the literature for UC. One of the first papers proposing new valid inequalities for the *3-bin* formulation of UC was **?**. This paper was very influential, and several subsequent papers improved most of the results presented there. In particular, **?** presented several types of new constraints based on ramp limits. Then **?** proposed new inequalities based on the study of two distinct polytopes which include the UC polytope: the ramp-up and ramp-down polytopes (the ramp-up polytope is the special case of 1UC obtained by relaxing ramp-down and shut-down limits, i.e., $\Delta_i^- = \bar{u}_i = \infty$, the ramp-down polytope is defined symmetrically); subsequently, **?** improved some of these result by considering the full UC polytope. In **?** the special case where only start-up and shut-down limits are imposed were characterized. On a different line of research, **??** analyzed the start-up costs: the former paper presented the case where only "cooling" (i.e., no "banking") is allowed, while the latter gave a complete polyhedral characterization for the definition of general history-dependent start-up costs. Finally, **?** presented a comprehensive review of the previous results and some new type of inequalities, studied the ways to combine the different types of inequalities in new models, and presented a large computational experience. All the above results show that describing the convex hull of (1UC) solutions, when all the technical constraints are considered, is highly nontrivial. Hence, formulations used in practice usually have to resort to carefully picking only some of the above ideas.

4. The dynamic programming algorithm

While (1UC) is nontrivial to describe in the variable spaces proposed in the previous sections, it is actually relatively easy to solve. Indeed, in **?** a Dynamic Programming (DP) algorithm was proposed that can solve (1UC) with all the constraints—minimum up- and down-time, ramp and generation limits—in $O(n^3)$ with the standard quadratic separable cost function (**??**) (and that can be generalized to more complex objectives). We now present an improvement of that DP algorithm and recall the basic ingredients of the approach that are necessary to present the MIP formulation.

In this paragraph, since the unit index $i \in I$ is fixed we will drop it for notational simplicity. The DP algorithm is based on defining a state-space graph $G = (N, A)$. The nodes in N are of two types:

ON_t and OFF_t for each $t \in T$, plus two special nodes, the source s and the sink d. The arcs in A are of two types: arcs (OFF_h, ON_k), denoting that the unit is turned ON at the beginning of time period h and that the unit remains ON until the end of time period k (indicated as *ON arcs*); arcs (ON_k, OFF_r), denoting that the unit is OFF from time periods $k+1$ to time period $r-1$ (indicated as *OFF arcs*). The OFF arcs satisfy minimum-down time constraints, that is $(ON_k, OFF_r) \in A$ if and only if $r \geq k + \tau^- + 1$, and are labeled with the start-up cost, here denoted by $\tilde{c}_-^{kr}$, that depends on the length $r - k - 1$ of the off period. Note that the most general time-dependent start-up costs (**??**) are easily handled within this framework, since the computation is done entirely offline. The ON arcs satisfy minimum-up time constraints, that is $(OFF_h, ON_k) \in A$ if and only if $k \geq h + \tau^+ - 1$, and are labeled with the cost of the optimal dispatch in the associated period. This is composed of two parts: fixed cost and variable cost. The first is just $\tilde{c}_+^{hk} = (k - h + 1)c$ since the unit will be committed in the interval $T(h,k)$. The variable cost, that depends on the p_t variables, is the optimal value of the following *Economic Dispatch problem with Ramping Constraints*

$$(ED^{hk}) \qquad \begin{aligned} \min\ & \textstyle\sum_{t \in T(h,k)} f(p_t) \\ & l \leq p_t \leq u \qquad t \in T(h,k) \\ & p_h \leq \bar{l} \\ & p_{t+1} \leq p_t + \Delta^+ \quad t \in T(h,k-1) \\ & p_t \leq p_{t+1} + \Delta^- \quad t \in T(h,k-1) \\ & p_k \leq \bar{u} \end{aligned}$$

Since all the relevant binary variables are fixed, (ED^{hk}) is an optimization problem with convex objective function and linear constraints. Hence, its optimal objective function value $z^{hk} = z(ED^{hk})$ can be computed in polynomial time.

Moreover, there are the connections between the source node s and the ON and OFF nodes defined according to the initial state of the unit. That is, if the unit is committed since τ^0 time periods, then there is an arc from s to each node ON_k such that $k + \tau^0 \geq \tau^+$. If, instead, the unit is uncommitted since $-\tau^0$ time periods, then there is an arc from s to each node OFF_h such that

$h - \tau^0 - 1 \geq \tau^-$; the latter arcs are labeled with the corresponding start-up cost. The ending node defines the type of the arcs starting from s. All nodes are then connected to the sink node d: arcs (ON_t, d) are of type OFF, arcs (OFF_t, d) are of type ON and their costs is computed for the period $T(t, n)$ only. The arc (s, d) means that the unit remains with the same status as at the beginning of the period and it is an ON arc if the unit was ON at time 0, and an OFF arc with zero cost if the unit was OFF at time 0.

Summing up, the state-space graph G has $2n + 2$ nodes and $O(n^2)$ arcs; every s–d path on G represents a feasible schedule for the unit. Hence, (1UC) is reduced to a shortest path problem on an acyclic graph with $O(n)$ nodes and $O(n^2)$ arcs. Thus, the problem can be solved in $O(n^2)$ once all the data has been computed. We remark that a larger, more complex graph with $O(n^2)$ nodes was proposed in **?**, but the one described in this paragraph (that appeared in **?**) is clearly preferable. Yet, in **?** it is proved that all $O(n^2)$ Economic Dispatch problems with Ramping Constraints can be solved in $O(n^3)$ by means of another Dynamic Programming algorithm, which is therefore the cost of the overall procedure, as it was with the original graph. The new graph state-space graph G will be the starting point for developing our MIP formulation in next paragraph.

Some different types of improvements have been obtained with respect to the above algorithm in different papers. In **?** the DP algorithm was improved by computing only the cumulative functions that are relevant at each time stamp. In **?** a similar algorithm was presented that has the advantage to works also for non-convex piece-wise linear cost functions. Finally, **?** introduced an algorithm to solve 1UC in $O(n)$ time with convex piece-wise linear generation cost and with ramp up and ramp down limits equal to each other.

5. The convex hull for the thermal single-unit polytope

In this section we introduce a new formulation for (1UC) that is inspired by the DP algorithm presented in Section **??**. This new formulation is composed of two parts:

- the shortest path formulation based on the state-space graph G of the DP algorithm;
- new power variables, their related cost, and the linking constraints with the previous part.

As in the previous section, the unit index $i \in I$ is fixed and therefore we drop it.

The shortest path formulation is straightforward: one just introduces the node-arc incidence matrix of the graph and writes the obvious system of inequalities. Then we can then simply write this part of the formulation as

$$Ey = \delta \ , \ y \geq 0 \ , \tag{18}$$

where E is the node-arcs incidence matrix of G, y is the vector of arc flow variables, and δ is the vector with all zero entries except $\delta_s = -1$ and $\delta_d = 1$ for the source node s and the sink node d, respectively. Hereafter, we denote with $A_{ON} = \{(h,k) | h,k \in T, (OFF_h, ON_k) \in A\}$ the set of pairs (h,k) such that (OFF_h, ON_k) is an ON arc (and symmetrically $A_{OFF} = \{(k,r) | k,r \in T, (ON_k, OFF_r) \in A\}$). The vector y will be partitioned according to arcs in A_{ON} and A_{OFF} as follows: y_+^{hk} for each $(h,k) \in A_{ON}$ and y_-^{kr} for each $(k,r) \in A_-$.

We now add variables p_t^{hk} associated with each ON arc $(OFF_h, ON_k) \in A$ and with $t \in T(h,k)$ to compute the power level for each time instant and the related costs. With these we define the *Extended Economic Dispatch* (with Ramping Constraints) sub-problem

$$(EED^{hk}) \qquad \begin{array}{l} \min \ \tilde{c}_+^{hk} y_+^{hk} + \sum_{t \in T(h,k)} f(p_t^{hk}) \\ \left. \begin{array}{ll} l y_+^{hk} \leq p_h^{hk} \leq \bar{l} y_+^{hk} & \\ l y_+^{hk} \leq p_t^{hk} \leq u y_+^{hk} & t \in T(h+1, k-1) \\ l y_+^{hk} \leq p_k^{hk} \leq \bar{u} y_+^{hk} & \\ p_{t+1}^{hk} \leq p_t^{hk} + y_+^{hk} \Delta^+ & t \in T(h, k-1) \\ p_t^{hk} \leq p_{t+1}^{hk} + y_+^{hk} \Delta^- & t \in T(h, k-1) \end{array} \right\} \\ y_+^{hk} \in \{0,1\} \end{array} \tag{19}$$

Basically, this is the Economic Dispatch (ED^{hk}) corresponding to traversing the arc $(OFF_h, ON_k) \in A$. It is easy to describe the convex hull of (EED^{hk}) due to the fact that, together with the *single* variable y_+^{hk} representing the traversal of the arc, it has a "private copy" of all involved continuous variables that are semi-continuous and all "governed" by the *same* y_+^{hk}. Therefore, the already recalled results about the Perspective Reformulation show that all that is needed for this is to replace the objective in (EED^{hk}) with

$$\min\ \tilde{c}_+^{hk} y_+^{hk} + \left(h(p^{hk}, y_+^{hk}) = \sum_{t \in T(h,k)} y_+^{hk} f(p_t^{hk} / y_+^{hk}) \right)\ . \tag{20}$$

Hence, the following convex NLP

$$\begin{aligned} \min\ & \tilde{c}_+^{hk} y_+^{hk} + \textstyle\sum_{t \in T(h,k)} z_t^{hk} \\ & z_t^{hk} \geq y_+^{hk} f(p_t^{hk} / y_+^{hk}) \quad t \in T(h,k) \\ & (\textbf{??})\ ,\ y_+^{hk} \in [0,1] \end{aligned} \tag{21}$$

is equivalent to (EED^{hk}), i.e., its constraint set describes the convex hull of the feasible solutions of (EED^{hk}). Note that in (**??**) we have made the objective linear with the well-known reformulation trick of introducing the auxiliary variables z_t^{hk} and moving the nonlinear part of the objective function into the constraints that define them. This puts the problem in the form required by our results below.

Separately, (**??**) has the integrality property, and therefore it would represent the convex hull of (1UC) were the objective function linear. We will show that the combination of (**??**) and (**??**) preserves the property, i.e., defines the convex hull of (1UC), using the nonlinear analogous of the well-known "Approach no. 4" of **?** (used by **?** and by others). To do that, we need the following characterization of the (closed) convex hull of a Mixed-Integer convex nonlinear set (**?**, §V.2):

Proposition 1. *Consider the closed convex NLP set*

$$C = \{\, z \in \mathbb{R}^n \,:\, f(z) \leq 0 \,\}\ ,$$

where $f : \mathbb{R}^n \to \mathbb{R}^m$*, and its mixed-integer restriction*

$$S = \{\, z \in C \,:\, z_k \in \mathbb{Z} \quad k \in K \subseteq \{\, 1, \ldots, n \,\} \,\}\ .$$

For any arbitrary objective function $c \in \mathbb{R}^n$*, let*

$$\sigma_C(c) = \inf\{\, cz \,:\, z \in C \,\} \geq D(c) = \sup_{\lambda \geq 0} \{\, L(\lambda; c) = \inf\{\, cz + \lambda f(z) \,\} \,\}\ , \tag{22}$$

(the minimization of cz over C, a.k.a. the support function *of C, and its Lagrangian Dual, $L(\lambda; c)$ being the Lagrangian function). If the condition*

$$\forall\, c \in \mathbb{R}^n \;\; \sigma_S(c) = \inf\{\, cz \,:\, z \in S \,\} = D(c) \tag{23}$$

holds, then $C = \overline{conv}(S)$, where $\overline{conv}$ denotes the closure of the convex hull.

We remark that, in Proposition **??**, $\sigma_S(c)$ and $D(c)$ need not be finite-valued; in particular, $\sigma_S(c) = -\infty$ may happen if S is not compact, which is the case in our application (as (**??**) is an epigraphical set, and therefore "unbounded from above"). However, $\sigma_S(c) = -\infty$ immediately implies $D(c) = -\infty$ (via $\sigma_C \leq \sigma_S$ and weak duality), i.e., $L(\cdot; c) = -\infty$ uniformly. What is needed is therefore that $D(c) = \sigma_S(c)$ when $\sigma_S(c)$ is finite. In turn, a necessary (but not sufficient) condition for this to happen is strong duality in the relaxation, i.e., $D(c) = \sigma_C(c)$, which typically requires some standard constraint qualification to hold.

The required result is now that, for an appropriate definition of *composition* of MINLP sets, the description of the convex hull of the composed set can be obtained from the descriptions of the convex hulls of the composing ones.

Definition 1. For $h = 1, 2$, let $S^h \subset \mathbb{R}^{n_h} \times \mathbb{R}$ be two sets; their *1-sum composition* is

$$S^1 \oplus S^2 = \{\, (x^1, x^2, y) \in \mathbb{R}^{n_1+n_2+1} \,:\, (x^h, y) \in S^h \quad h = 1, 2 \,\} \;.$$

For future reference, let us remark that 1-sum composition preserves both convexity and closedness. Indeed, $S^1 \oplus S^2$ is isomorphic to the set $(S^1 \times \mathbb{R}^{n_2}) \cap (\mathbb{R}^{n_1} \times \tilde{S}^2)$, where $\tilde{S}^2 = \{(y, x^2) | (x^2, y) \in S^2\}$, and both Cartesian product and intersection separately preserve both convexity and closedness. Since the result hinges on duality, some mild requirements are necessary on the algebraic representation of the convex hulls (cf. f in Proposition **??**). To keep the result as general as possible, we will state them in the most abstract way possible:

Assumption 1. *For each (closed convex) set C represented by constraint (closed convex) functions $f = [\, f_i \,]_{i=1,\dots,m} : \mathbb{R}^n \to \mathbb{R}^m$, assumptions hold such that the KKT conditions of* (**??**)

$$\exists\, \lambda \in \mathbb{R}^m_+ \; s.t. \; 0 \in c + \textstyle\sum_{i=1}^m \lambda_i \partial f_i(x) \;\;,\;\; f(x) \leq 0 \;\;,\;\; \lambda f(x) = 0 \tag{24}$$

(?, Theorem VII.2.1.4) are both necessary and sufficient for global optimality of x, where $\partial f_i(x)$ denotes the standard subdifferential of f_i at x (?, Chap. VI).

These are mild conditions in practice. For instance, for convex f_i (by far the most common occurrence) several classical constraint qualifications, like Slater, linearly independence and affinity, suffice. In general, a convex set can be represented also by nonconvex functions; in **?**, for instance, the Slater condition plus a nondegeneracy one ($\nabla f_i(x) \neq 0$ whenever $f_i(x) = 0$ and $x \in C$) is shown to suffice as well. Thus, the conditions are typically satisfied by standard MINLP models, such as our (**??**) and (**??**).

Lemma 1. *For $h = 1, 2$, let $S^h \subset \mathbb{R}^{n_h} \times \mathbb{R}$ be two sets. If: i) the closed (convex) sets*

$$C^h = \left\{ (x^h, y) \in \mathbb{R}^{n_h+1} : y \geq 0 \ , \ f^h(x^h, y) \leq 0 \right\} \tag{25}$$

describe the convex hull of S^h, ii) Assumption **??** *holds, iii) $(x^h, y) \in S^h$ implies that $y \in \{0, 1\}$ and, iv) there exist points $(\bar{x}^h, 0) \in S^h$ and $(\tilde{x}^h, 1) \in S^h$, for $h = 1, 2$, then $C^1 \oplus C^2 = \overline{conv}(S^1 \oplus S^2)$.*

The proof of Lemma **??** is rather long and technical, therefore we report it in the Appendix.

Note that assumption (iv) corresponds to the "complex" case that requires proof, but the Theorem clearly holds in general. Indeed, if, say, there are no solutions $(x^1, 0)$ and $(x^1, 1)$ then $S^1 = C^1 = \emptyset$; hence, $C^1 \oplus C^2 = S^1 \oplus S^2 = \overline{conv}(S^1 \oplus S^2) = \emptyset$. The same holds if, say, there exist points $(x^1, 0)$ and $(x^2, 1)$ but not $(x^1, 1)$ and $(x^2, 0)$. Finally, if, say, there exist points $(x^1, 0)$ and $(x^2, 0)$ but not $(x^1, 1)$ and $(x^2, 1)$, then S^1 and S^2 are convex and therefore the result trivially holds.

We are now ready for the announced result:

Theorem 1. *The formulation*

$$\begin{aligned} & min\ \tilde{c}y + \textstyle\sum_{(h,k) \in A_{ON}} z^{hk} \\ & \quad (\mathbf{??}) \\ & \quad z^{hk} \geq \textstyle\sum_{t \in T(h,k)} y_+^{hk} f(p_t^{hk} / y_+^{hk}) \ \ (h,k) \in A_{ON} \ , \\ & \quad (\mathbf{??}) \qquad\qquad\qquad\qquad\qquad\qquad (h,k) \in A_{ON} , \end{aligned} \tag{26}$$

where $\tilde{c} = [[\tilde{c}_+^{hk}]_{(h,k)\in A_{ON}}, [\tilde{c}_-^{kr}]_{kr\in A_{OFF}}]$ *is the vector of all arc costs and* $y = [[y_+^{hk}]_{(h,k)\in A_{ON}}, [y_-^{kr}]_{(k,r)\in A_{OFF}}]$, *describes the convex hull of the feasible solutions for (1UC).*

Proof. Define S_0 the set of feasible solutions of the network flow problem (**??**) associated with the DP graph G, and S^{hk} the set of feasible solutions of (**??**). We can build the set of solutions for the complete problem by iteratively composing the solutions of S_0 with the sets S^{hk}, e.g., in lexicographic order of the pairs $(h,k) \in A_{ON}$. At each step $j \geq 1$ of the process, we are combining a set S_{j-1} and a set S^{hk} (for some fixed pair (h,k)) that only share the single binary variable y_+^{hk} to obtain the set $S_j = S_{j-1} \oplus S^{hk}$. It is immediate to prove by induction that the convex NLP formulation obtained by adding to the inequalities of the system (**??**) all the inequalities of (**??**) for all the pairs (h,k) used describes the convex hull of S_j, and satisfies strong duality for each objective function. Indeed, at the first iteration we are combining S_0 with one S^{hk}; both sets satisfy the hypotheses of Lemma **??** (having only linear constraints, S_0 does not need any strict feasibility assumption for strong duality to hold), and therefore also the corresponding formulation satisfies them for $S_1 = S_0 \oplus S^{hk}$. Repeating the process, at each step the corresponding convex NLP describes $conv(S_j)$ and satisfies strong duality. At the end of the composition process we have obtained all the constraints in (**??**), which therefore define a convex NLP formulation for the convex hull of the overall set of solutions for (1UC). □

Formulation (**??**) is the first formulation for (1UC) that describes the convex hull of feasible solutions considering all the constraints described above and a convex objective function. As previously mentioned, the formulation presented in **?** was claimed to have this property, but the claim is proven false in **?** with a counterexample.

Finally, we present a formulation for the UC problem based on the exact single-unit formulation can then be summarized as follows:

$$
\begin{aligned}
\min\ & \textstyle\sum_{i\in I} \tilde{c}_i y_i + \sum_{i\in I} \sum_{hk\in A_{ON}} z_i^{hk} \\
& \textstyle\sum_{i\in I} \sum_{hk:t\in T(h,k)} p_{it}^{hk} = d_t \qquad t \in T \\
& \textstyle z_i^{hk} \geq \sum_{t\in T(h,k)} y_{i+}^{hk} f^i(p_{it}^{hk}/y_{i+}^{hk}) \quad (h,k)\in A_{ON}^i\ ,\, i\in I \\
& (\mathbf{??})\ ,\ (\mathbf{??}) \qquad i \in I.
\end{aligned}
\tag{27}
$$

In the rest of the paper, we will refer to (**??**) as the *DP formulation*. Note that the number of variables is $O(n^2)$ (n being the number of time instants) for the network flow system and $O(n)$ for each of the $O(n^2)$ subproblems (**??**) associated with each pair (h,k); hence, the total number of variables in (**??**) is $O(n^3)$.

6. Additional DP based formulations

We now introduce two other formulations based on the ideas developed in Section **??** and that attain different trade-offs between size and quality of the continuous relaxation lower bound. When restricted to (1UC), both are less tight than the exact formulation (**??**) but keep the network constraints (**??**). The first one uses, as in the model T by **?**, the power variables p'_{it}, while the second one presents a new type of variables whose cardinality is intermediate between *3-bin* and DP formulations.

6.1. The p_t model

Given a unit i, consider the commitment variable x_{it}, the start-up/shut-down variables v_{it}/w_{it} and the set of variables y_{i+}^{hk} associated with ON arcs (OFF_h, ON_k) such that $t \in T(h,k)$. It is easy to see that, by definition, these variables are related by the following equations:

$$x_{it} = \sum_{(h,k):t\in T(h,k)} y_{i+}^{hk} \ , \ v_{it} = \sum_{k\geq t} y_{i+}^{tk} \ , \ w_{it+1} = \sum_{h\leq t} y_{i+}^{ht} \ . \tag{28}$$

Consequently, considering that $p_{it} = p'_{it} + l_i x_{it} = p'_{it} + l_i \sum_{(h,k):t\in T(h,k)} y_{i+}^{hk}$, the ramp-up/ramp-down constraints assume, respectively, the following form:

$$p'_{it} - p'_{it-1} \leq \Delta_i^+ \sum_{(h,k):t-1\in T(h,k-1)} y_{i+}^{hk} + (\bar{l}_i - l_i) \sum_{k:k\geq t} y_{i+}^{tk} \qquad i\in I, t\in T(2,n) \tag{29}$$

$$p'_{it-1} - p'_{it} \leq \Delta_i^- \sum_{(h,k):t-1\in T(h,k-1)} y_{i+}^{hk} + (\bar{u}_i - l_i) \sum_{h:h\leq t-1} y_{i+}^{ht-1} \qquad i\in I, t\in T(2,n) \tag{30}$$

Note that, in case the unit is on at the beginning of time horizon ($\tau_i^0 > 0$), the initial ramp-up/ramp-down conditions have to be set by

$$p'_{i1} \leq (\Delta_i^+ + p_{i0} - l_i) \sum_{k:1\leq k} y_{i+}^{0k} \qquad i\in I : \tau_i^0 > 0 \tag{31}$$

$$-p'_{i1} \leq (\Delta_i^- - p_{i0} + l_i) \sum_{k:1\leq k} y_{i+}^{0k} \qquad i\in I : \tau_i^0 > 0 \tag{32}$$

Then, the minimum and maximum power output constraints can be re-written as follows:

$$p'_{it} \le (u_i - l_i) \sum_{(h,k): t \in T(h,k)} y^{hk}_{i+} \qquad i \in I, t \in T \tag{33}$$

The right-hand side of constraints (**??**) can be reinforced as follows. Assuming that $\tau^+_i \ge 2$, if a unit i is switched on at time t then $\sum_{k:k\ge t} y^{tk}_{i+} = 1$ and the power p'_{it} is bounded by $\bar{l}_i - l_i$. If the unit is switched off at time t then $\sum_{h:h\le t} y^{ht}_{i+} = 1$ and the power p'_{it} does not exceed $\bar{u}_i - l_i$. In case the unit does not turn on or off but it is committed at time t then $\sum_{(h,k):h<t<k} y^{hk}_{i+} = 1$ holds. Consequently, there exists (h,k) such that $h < t < k$ and $y^{hk}_{i+} = 1$. Therefore, because of the maximum power output and the ramp-up/ramp-down constraints, the power p'_{it} is bounded by $\psi'^{hk}_{it} = \min\{u_i - l_i, \bar{l}_i - l_i + \Delta^+_i(t-h), \bar{u}_i - l_i + \Delta^-_i(k-t)\}$. Furthermore, if the unit is initially committed ($\tau^0_i > 0$) then $\sum_{k:1\le k} y^{0k}_{i+} = 1$ and we have to set $\psi'^{0k}_{it} = \min\{u_i - l_i, p_{i0} - l_i + \Delta^+_i \cdot t, \bar{u}_i - l_i + \Delta^-_i(k-t)\}$. Note that, if $\tau^+_i \ge 2$, the variable y^{tt}_{i+} is not defined. Then, if $\tau^+_i = 1$ and $y^{tt}_{i+} = 1$, the power p'_{it} is bounded by the minimum between $\bar{l}_i - l_i$ and $\bar{u}_i - l_i$.

Hence, if $\tau^+_i \ge 2$, the right-hand side of constraints (**??**) can be reinforced as

$$p'_{it} \le (\bar{l}_i - l_i) \sum_{k:k\ge t} y^{tk}_{i+} + (\bar{u}_i - l_i) \sum_{h:h\le t} y^{ht}_{i+} + \sum_{(h,k):h<t<k} \psi'^{hk}_{it} y^{hk}_{i+} \qquad i \in I : \tau^+_i \ge 2, t \in T \tag{34}$$

Otherwise, if $\tau^+_i = 1$, this can be done as follows, for all $i \in I$ and $t \in T$:

$$p'_{it} \le (\bar{l}_i - l_i) \sum_{k:k>t} y^{tk}_{i+} + (\bar{u}_i - l_i) \sum_{h:h<t} y^{ht}_{i+} + \sum_{(h,k):h<t<k} \psi'^{hk}_{it} y^{hk}_{i+} + (\min\{\bar{l}_i, \bar{u}_i\} - l_i) y^{tt}_{i+} \tag{35}$$

Then, the demand constraints (**??**) becomes

$$\sum_{i\in I} (p'_{it} + l_i \sum_{(h,k):t\in T(h,k)} y^{hk}_{i+}) = d_t \qquad t \in T \tag{36}$$

Finally, the objective function is

$$\min \sum_{i\in I} \tilde{c}_i y_i + \sum_{i\in I} \sum_{t\in T} (\sum_{(h,k):t\in T(h,k)} y^{hk}_{i+}) f^i((p'_{it} + l_i \sum_{(h,k):t\in T(h,k)} y^{hk}_{i+}) / (\sum_{(h,k):t\in T(h,k)} y^{hk}_{i+})) \ . \tag{37}$$

We will denote as p_t-*model* the following formulation:

$$\min \big\{ (\mathbf{??}) : (\mathbf{??}) \ , (\mathbf{??})\text{–}(\mathbf{??}) \big\} \tag{38}$$

6.2. The Start-Up model

To introduce the last formulation, we define a variable p_{it}^{h} denoting the unit i is committed at time t and it has been turned on at time instant h. Differently from the variable p_{it}^{hk}, with p_{it}^{h} the time when the unit will be turned off is not specified. The relation between variables p_{it} and p_{it}^{h} is

$$p_{it} = \sum_{h:h\leq t} p_{it}^{h} \tag{39}$$

In what follows, we present a new formulation based on the DP algorithm and by considering equation (**??**). In particular, the ramp-up/ramp-down constraints are the following ones:

$$p_{it}^{h} - p_{it-1}^{h} \leq -l_i y_{i+}^{ht-1} + \Delta_i^{+} \sum_{k:k\geq t} y_{i+}^{hk} \qquad i\in I, h\in T(1,n-1), t\in T(h+1,n) \tag{40}$$

$$p_{it-1}^{h} - p_{it}^{h} \leq \bar{u}_i y_{i+}^{ht-1} + \Delta_i^{-} \sum_{k:k\geq t} y_{i+}^{hk} \qquad i\in I, h\in T(1,n-1), t\in T(h+1,n) \tag{41}$$

Again, if $\tau_i^0 > 0$, the initial ramp-up/ramp-down conditions has to be imposed as

$$p_{i1}^{0} \leq (\Delta^{+} + p_0) \sum_{k:1\leq k} y_{i+}^{0k} \qquad i\in I : \tau_i^0 > 0 \tag{42}$$

$$-p_{i1}^{0} \leq (\Delta^{-} - p_0) \sum_{k:1\leq k} y_{i+}^{0k} \qquad i\in I : \tau_i^0 > 0 \tag{43}$$

Then, the minimum/maximum power output constraints are

$$l_i \sum_{k:k\geq t} y_{i+}^{hk} \leq p_{it}^{h} \leq u_i \sum_{k:k\geq t} y_{i+}^{hk} \qquad i\in I, h\in T(0,n), t\in T(h,n) \tag{44}$$

Note that when $t = h$, we can improve the right hand side of constraints (**??**) with

$$p_{ih}^{h} \leq \bar{l}_i \sum_{k:k>h} y_{i+}^{hk} + \min\{\bar{l}_i, \bar{u}_i\} y_{i+}^{hh} \qquad i\in I, h\in T \tag{45}$$

On the other hand, when $t > h$, constraints (**??**) can be improved by considering that the unit could be switched off at time t ($y_{i+}^{ht} = 1$) or not ($\sum_{k:k>t} y_{i+}^{hk} = 1$). In the former case the power p_{it}^{h} does not exceed $\bar{u}_i$. In the latter case, due to the maximum power output and ramp up/ramp-down constraints, the power p_{it}^{h} is bounded by $\sum_{k:k>t} \psi_{it}^{hk} y_{i+}^{hk}$, with $\psi_{it}^{hk} = \min\{u_i, \bar{l}_i + \Delta_i^{+}(t-h), \bar{u}_i + \Delta_i^{-}(k-t)\}$. Then, when $t > h$, the right hand side of constraints (**??**) can be enforced with

$$p_{it}^{h} \leq \bar{u}_i y_{i+}^{ht} + \sum_{k:k>t} \psi_{it}^{hk} y_{i+}^{hk} \qquad i\in I, h\in T(0,n-1), t\in T(h+1,n) \tag{46}$$

In conclusion, we derive the objective function and demand constrains

$$\min \sum_{i\in I} \tilde{c}_i y_i + \sum_{i\in I} \sum_{t\in T} \sum_{h:t\geq h} (\sum_{k:k\geq t} y_{i+}^{hk}) f^i(p_{it}^h / (\sum_{k:k\geq t} y_{i+}^{hk})) \tag{47}$$

$$\sum_{i\in I} \sum_{h:h\leq t} p_{it}^h = d_t \qquad t\in T \tag{48}$$

In the rest of the paper we will denote by *SU-model* the following

$$\min \{ (\mathbf{??}) : (\mathbf{??}) \; , \; (\mathbf{??}) \; , \; (\mathbf{??})\text{–}(\mathbf{??}) \} \tag{49}$$

7. Computational tests

In this section we test the computational performances of the new DP formulation presented in Section **??** and formulations p_t and *SU* introduced in Section **??**. The main issue, of course, is that of the trade-off between the bound improvement and the cost increase due to the larger size. Indeed, the *3-bin* formulation has $O(n)$ variables only and $O(n)$ constraints for each unit, while the new ones proposed have $O(n^2)$ binary variables, at least $O(n)$ continuous variables, and at least $O(n)$ constraints, for each unit. The new formulations mainly differ on the type of continuous variables and associated constraints. The *DP-model* (**??**) contains $O(n^3)$ variables p_{it}^{hk} for each unit i, the p_t*-model* (**??**) contains $O(n)$ variables p'_{it} for each unit i, and the *SU-model* (**??**) represents an intermediate case and contains $O(n^2)$ variables p_{it}^h for each unit i. Table **??** gives an overview of the size of each formulation for each unit.

The experiments have been carried out with CPLEX 12.10, setting one thread and a time limit of 10000 seconds on a PC with 2.2 GHz Intel Xeon Gold 5120 CPUs and 64 GB of RAM, under a GNU/Linux Ubuntu 20.04.1 LTS operating system. Except for the use of a single thread, any other setting was left by default. The code of the algoritms used in the computational experiments can be downloaded at

`https://gitlab.com/smspp`

Model	Power variables	Other variables	Constraints
3-*bin*	$O(n)$	$O(n)$	$O(n)$
DP	$O(n^3)$	$O(n^2)$	$O(n^3)$
p_t	$O(n)$	$O(n^2)$	$O(n^2)$
SU	$O(n^2)$	$O(n^2)$	$O(n^2)$

Table 1 Bounds on the number of variables and constraints for each model and each unit.

We used two sets of instances published at

`https://commalab.di.unipi.it/datasets/UC/`

considering pure thermal instances ranging from 10 to 54 units and $n = 24$ time periods. All the instances consider fixed start-up costs. Note that this choice is in favour of the *3-bin* formulation because history-dependent start-up costs can be added to DP-based formulations without additional computational cost, whereas they would both increase the size and weaken the bound of the *3-bin* one.

The first dataset, denoted as DS1, was generated by means of the UCIG generator available at

`http://groups.di.unipi.it/optimize/Data/UC/ucig.tgz`

The generator produces a generating set with "small", "medium" and "large" thermal units in realistic proportions; the characteristics of each unit are then randomly generated within a set of realistic parameters, depending on the type of the unit. Since the generator does not produce ramping data, we have generated at random for each unit i two real values $r_u, r_d \in [2, 3]$ such that $\Delta_i^+ = r_u(u_i - l_i)$ and $\Delta_i^- = r_d(u_i - l_i)$. In this way, we produced realistic conditions where a unit could reach the maximum power in two/three time intervals without excessive thermal stress. We generated five instances with 10, five with 20 and five with 50 thermal units and $n = 24$ time periods.

The second dataset, denoted here as DS2, includes two instances each with 54 thermal units and $n = 24$ time periods. More precisely, these two instances correspond to the well-known instance IEEE 118-Bus Test System and they differ in the values of the start-up costs, only. Starting from each instance of each dataset with $n = 24$, an instance with $n = 36$ was created by setting the same demand values for the first 24 time periods and randomly generating the demands d_t for each $t \in [25\,,\,36]$ such that d_t differs from the demand d_{t-24} by a maximum of five percent. Similarly, an instance $n = 48$ was generated from each one with $n = 36$ by maintaining the demands for the first 36 time periods and randomly generating d_t for $t \in [37\,,\,48]$ from d_{t-24} with an error of five percent. Finally, only for the instances of the dataset DS2, an instance with $n = 72$ was generated from each one with $n = 48$ by maintaining the demands for the first 48 time periods and randomly generating d_t for $t \in [49\,,\,72]$ from d_{t-24} with an error of five percent.

We considered in total 45 instances in DS1 and 8 in DS2 and we present the average of the results thus obtained.

In Table **??** and Table **??** we compare the running times in seconds to solve the continuous relaxation (time^{lp}) of the new formulations presented in this paper and the standard *3-bin* one, with the corresponding gap in percentage (gap^{lp}%) w.r.t. the best integer solution ever found among all formulations. Note that the gap is the one of the "pure" formulation, i.e., before any cut added by CPLEX. The results of dataset DS1 in Table **??** show that DP-model always provides the best gaps and the *3-bin* worst ones. However, the running time of the DP-model is orders of magnitude larger, up to the point that for larger sizes it cannot be computed within the time limit of 10000 seconds for at least one instance (cells with symbol "−"). On the other hand, the p_t-model and SU-model provide better gaps than the *3-bin* one within reasonable computing times.

The same happens on dataset DS2 (Table **??**) although these instances are more difficult for the formulation DP that does not provide any results for n grater than 24. When available, the best gaps are given by the DP formulation, otherwise by SU (that does not give results for $n = 72$) and p_t. The worst gaps are also in this dataset those of *3-bin*.

We have also tested solving the UC problem at integer optimality, with gap 0.01% (Table **??** for dataset DS1 and Table **??** for dataset DS2) and with gap 0.1% (Table **??** for dataset DS1). In the tables, for each model, column *time*(*opt*) reports the average computing time in seconds and the number of instances, over five in DS1 and over two in DS2, solved to optimality (in parentheses) within the time limit. Columns *nodes* and *gap*% denote the average number of nodes explored and the average final gap, respectively.

Overall, the results in Table **??** show that the p_t-model is competitive. The *3-bin* one is faster on smaller number of units, instances, but the p_t-model most often than not solves more instances, or is faster, or yields a better final gap. As p_t-model wins over the *3-bin* model for different cases ($n = 24$ and $|I| = 50$, $n = 36$ and $|I| = 20$, $n = 48$ and $|I| = 50$), it can be safely stated that the p_t-model has at least very close performances to the *3-bin* one.

Obviously, by setting a lower gap (Table **??**) more instances are solved. This makes *3-bin* more competitive, as it dominates the p_t-model for all $n = 24$; but even the p_t-model then always solves all instances in less than half a minute. As the size grows the picture gets more complex: the p_t-model is competitive (6% worse in CPU time) on 36-50 instances and it solves one instance more on the most difficult 48-50, although suffering a surprising setback on 48-20 ones.

The results in Table **??** show that overall the p_t formulation has better results than *3-bin*. In fact, for $n = 24$ and $n = 36$ both formulations solve all the instances, but the p_t has much lower computing times. For $n = 48$, where no instance is resolved, the p_t formulation provides better gaps than those of *3-bin*. Only for $n = 72$ the *3-bin* is better than the p_t giving lower gaps. The DP and SU models, on the other hand, are decidedly less performing than the *3-bin* and p_t formulations.

All in all, these results show that the DP-based formulations, and in particular the p_t-model, are promising for the computational solution of large, hard UC instances. This is true in particular due to the fact that they lend themselves very well to the *Stabilised Structured Dantzig-Wolfe* technique **?**, that has the potential of significantly reducing the size of the continuous relaxations solved by incrementally generating them piecemeal. Inserting this process in a full-fledged Branch-and-Cut approach without losing all the knowledge and machinery already integrated in present

MIP solvers is not at all straightforward. However, we believe it has a significant potential to yield more effective solution approaches to UC. This issue will be investigated in a future paper.

		3-*bin*		*DP*		p_t		*SU*	
n	units	time^{lp}	gap^{lp}%	time^{lp}	gap^{lp}%	time^{lp}	gap^{lp}%	time^{lp}	gap^{lp}%
	10	0.10	1.01	18.77	0.67	0.20	0.78	3.25	0.78
24	20	0.29	0.82	69.68	0.51	0.59	0.52	9.92	0.51
	50	1.31	0.32	310.69	0.08	2.41	0.08	25.02	0.08
	10	0.22	0.96	224.00	0.69	0.66	0.78	15.89	0.77
36	20	0.66	0.69	910.95	0.44	2.27	0.45	40.90	0.44
	50	3.55	0.29	5108.07	0.11	10.75	0.12	131.68	0.11
	10	0.33	0.98	1667.43	0.74	1.77	0.81	48.81	0.80
48	20	1.13	0.67	–	–	5.64	0.44	130.55	0.45
	50	5.97	0.29	–	–	30.51	0.11	449.66	0.15

Table 2 Root node gaps of the *DP*, p_t, *SU* and 3-*bin* formulations on dataset DS1

		3-*bin*		*DP*		p_t		*SU*	
n	units	time^{lp}	gap^{lp}%	time^{lp}	gap^{lp}%	time^{lp}	gap^{lp}%	time^{lp}	gap^{lp}%
24	54	3.40	0.22	3659.58	0.10	6.64	0.12	50.06	0.11
36	54	3.74	0.45	–	–	25.67	0.14	537.14	0.12
48	54	6.33	0.44	–	–	62.91	0.14	1873.59	0.13
72	54	8.12	0.49	–	–	212.23	0.15	–	–

Table 3 Root node gaps of the *DP*, p_t, *SU* and 3-*bin* formulations on dataset DS2

		3-bin			*DP*			p_t			*SU*		
n	units	time(opt)	nodes	gap%	time(opt)	nodes	gap%	time(opt)	nodes	gap%	time(opt)	nodes	gap%
	10	2(5)	23	0.01	653(5)	610	0.01	5(5)	46	0.01	99(5)	437	0.01
24	20	107(5)	535	0.01	8082(2)	1789	0.06	377(5)	932	0.01	2754(4)	1885	0.01
	50	7678(2)	1959	0.02	9699(1)	655	0.07	5089(3)	1585	0.02	8078(1)	1777	0.05
	10	17(5)	181	0.01	10000(0)	785	0.48	98(5)	537	0.01	4995(4)	2643	0.04
36	20	8997(1)	2362	0.03	10000(0)	68	–	8044(1)	1912	0.06	10000(0)	1630	0.22
	50	10000(0)	1204	0.05	10000(0)	2	–	10000(0)	1096	0.07	10000(0)	719	0.28
	10	987(5)	894	0.01	10000(0)	8	–	5072(3)	2334	0.04	10000(0)	1364	0.73
48	20	10000(0)	1297	0.10	10000(0)	0	–	10000(0)	1302	0.14	10000(0)	840	0.43
	50	10000(0)	800	0.11	10000(0)	0	–	10000(0)	905	0.08	10000(0)	307	0.43

Table 4 Computational results with gap 10^{-4} on dataset DS1

		3-bin			*DP*			p_t			*SU*		
n	units	time(opt)	nodes	gap%	time(opt)	nodes	gap%	time(opt)	nodes	gap%	time(opt)	nodes	gap%
	10	1(5)	8	0.07	636(5)	501	0.07	3(5)	24	0.08	87(5)	296	0.10
24	20	1(5)	0	0.09	6699(4)	1320	0.10	45(5)	118	0.09	921(5)	823	0.10
	50	10(5)	6	0.07	5710(3)	295	0.10	29(5)	26	0.08	3092(4)	674	0.10
	10	5(5)	18	0.08	10000(0)	821	0.45	47(5)	152	0.09	4144(4)	1999	0.11
36	20	400(5)	198	0.08	10000(0)	74	–	2964(5)	750	0.09	10001(0)	1632	0.24
	50	2457(4)	422	0.09	10000(0)	2	–	2617(4)	418	0.09	8461(1)	616	0.19
	10	170(5)	225	0.07	10000(0)	7	–	1535(5)	1116	0.10	10000(0)	1363	0.74
48	20	6003(3)	679	0.13	10000(0)	0	–	10000(0)	1087	0.14	10000(0)	825	0.40
	50	4253(2)	305	0.12	10000(0)	0	–	7023(3)	662	0.12	10000(0)	306	0.41

Table 5 Computational results with gap 10^{-3} on dataset DS1

		3-bin			*DP*			p_t			*SU*		
n	units	time(opt)	nodes	gap%	time(opt)	nodes	gap%	time(opt)	nodes	gap%	time(opt)	nodes	gap%
24	54	103(2)	670	0.01	7483(2)	930	0.01%	51(2)	118	0.01	744(2)	470	0.01
36	54	8620(2)	7542	0.01	10000(0)	0	-	646(2)	880	0.01	10000(0)	1678	0.11
48	54	10000(0)	2736	0.08	10000(0)	0	-	10000(0)	3043	0.02	10000(0)	336	0.23
72	54	10000(0)	2966	0.08	10000(0)	0	-	10000(0)	1290	0.16	10000(0)	0	–

Table 6 Computational results with gap 10^{-4} on dataset DS2

8. Conclusions

We have presented the first exact formulation for the (1UC) problem with convex cost function and all the main operational constraints proposed in the literature. The formulation is based on the combination of the flow formulation of the DP approach, which “has the integrality property for the constraints”, and of the perspective reformulation for the objective, which “has the integrality property for the objective”. We believe that this combined approach is interesting in its own right, and it could have other applications. In fact, the proof technique we have used is fairly general, as it applies to any MINLPs that can be obtained via 1-sum composition by fragments with the integrality property, and it is a significant contribution in itself.

While the proposed (1UC) DP-based formulation indeed produces the strongest lower bounds when used as the basis of a (UC) formulation, its large size makes it not practical. We have therefore proposed and analyzed two alternative formulations, based on partial variable aggregation, that offer different trade-offs between size and potential bound quality. The experimental results show that in particular the p_t formulation is competitive with the best previously known one.

We believe that the new formulations presented in this paper show promise, and warrant further investigation in at least two directions. The first is to help in the definition of heuristic algorithms that exploit the much smaller gap and use the better continuous solution to quickly produce feasible solutions with the quality required by practical applications (say, a gap smaller than

0.5%). The second is the fact that, like with all "large" formulations, the number of variables and constraints that are actually required to characterize at the optimal solution is a small fraction of the total number. Thus, generation of variables and constraints, such as the Stabilised Structured Dantzig-Wolfe Decomposition (**?**), could very considerably speed-up the overall performances of the algorithm, thereby overcoming the disadvantage related to the larger size of the proposed formulations and making them even more competitive with the *3-bin* one.

Appendix A: Proof of Lemma ??

Arbitrarily choose $(c^1, c^2, d) \in \mathbb{R}^{n_1+n_2+1}$ and consider the problem

$$L = \inf\{\, c^1x^1 + c^2x^2 + dy \,:\, (x^h, y) \in S^h \quad h = 1, 2\,\} \ , \tag{50}$$

its relaxation

$$\Pi = \inf\{\, c^1x^1 + c^2x^2 + dy \,:\, (x^h, y) \in C^h \quad h = 1, 2\,\} \ , \tag{51}$$

(since $S^h \subseteq C^h$), and the Lagrangian Dual of (**??**)

$$\Delta = \sup_{\mu \geq 0\,,\, \lambda^1 \geq 0\,,\, \lambda^2 \geq 0} \{\, L(\mu, \lambda^1, \lambda^2)\,\} \ ,$$

where

$$L(\mu, \lambda^1, \lambda^2) = \inf_{x^1\,,\, x^2\,,\, y \geq 0} \{\, c^1x^1 + c^2x^2 + (d - \mu)y + \lambda^1 f^1(x^1, y) + \lambda^2 f^2(x^2, y)\,\} \tag{52}$$

is the Lagrangian function. Note in particular the term "$-\mu y$" in (**??**) coming from the explicit dualisation of the single constraint $y \geq 0$, which is independently present in the definition of S^h/C^h for both $h = 1, 2$ (cf. (**??**)) but that clearly need not be replicated twice when the combined problems (**??**) is solved. It is apparent that $(\Delta \leq)\, \Pi \leq L$: we want to prove that $L = \Delta\, (= \Pi)$ which, via Proposition **??**, yields the desired result. This proof follows similar steps as the proof by **?** for composition of stable set polyhedra by clique-cutsets.

As already remarked, the case $L = -\infty$ is possible, but not challenging: by weak duality $\Delta = -\infty$ as well and the result is established. Hence, we can focus on the case where $\Pi \geq \Delta > -\infty$. Since the feasible region in (**??**), i.e., $C^1 \oplus C^2$, is closed and the objective is linear, the problem admits an optimal solution (x^1, x^2, y); due to Assumption **??**, such an optimal solution is characterised by the KKT conditions (**??**)

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \in \begin{bmatrix} c^1 \\ c^2 \\ d \end{bmatrix} + \mu \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} + \sum_{i=1}^{m_1} \lambda_i^1 \partial f_i^1(x^1, x^2, y) + \sum_{i=1}^{m_2} \lambda_i^2 \partial f_i^2(x^1, x^2, y) \tag{53a}$$

$$\mu y = 0 \tag{53b}$$

$$\lambda^1 f^1(x^1, y) = 0 \tag{53c}$$

$$\lambda^2 f^2(x^2, y) = 0 \tag{53d}$$

together with primal feasibility—$(x^1, x^2, y) \in C^1 \oplus C^2$—and dual feasibility—$\mu \geq 0$, $\lambda^1 \geq 0$, $\lambda^2 \geq 0$—are both necessary and *sufficient* for optimality of (x^1, x^2, y) and $(\mu, \lambda^1, \lambda^2)$. We remark that an immediate consequence of (**??**) holding, and in particular of the *complementary slackness conditions* (**??**)–(**??**), is that the objective value of the primal and of the dual solution coincide, and therefore

$$\Pi = c^1 x^1 + c^2 x^2 + dy = \Delta \ .$$

For $h = 1, 2$, but also for *fixed* values $y \in \{0, 1\}$, we now define the quantities

$$L^h_y = \inf\{ \, c^h x^h + dy \, : \, (x^h, y) \in C^h \, \} \ .$$

Note that, by assumption (iv), L^h_y is defined; moreover, $L^h_y > -\infty$. Indeed, assume this was not the case for, say, $h = 1$ and $y = 0$, and denote with C^1_0 the projection on the x^1 sub-space of the slice of C^1 where y is fixed to 0 (note that C^1_0 is convex, since both slicing and projection preserve convexity). Hence, $L^1_0 = -\infty$ would mean that there exists a direction $v \in rec(C^1_0)$ such that $c^1 v < 0$. But this would mean that $[\, v \, , \, 0 \, , \, 0 \,] \in rec(C^1 \oplus C^2)$, and the scalar product with the objective is $c^1 v < 0$, which contradicts $\Pi > -\infty$. We can then define $L_0 = L^1_0 + L^2_0$, which is the optimal value of the restriction of (**??**) corresponding to choosing $y = 0$, and, similarly, $L_1 = L^1_1 + L^2_1 - d$, which is the optimal value of rather choosing $y = 1$ (note the "$-d$" term, due to a "$+d$" term being present twice in both L^1_1 and L^2_1 separately). It is then plain to see that $L = \min\{ \, L_0 \, , \, L_1 \, \}$ is the optimal value of (**??**). We aim at constructing primal and dual optimal solutions for (**??**) (which, therefore, satisfy (**??**)) whose objective value is precisely L.

To do that, again for $h = 1, 2$ we define the auxiliary problems

$$\sigma^h = \inf\{ \, c^h x^h + (d + L^h_0 - L^h_1) y \, : \, (x^h, y) \in S^h \, \} \tag{54}$$

$$\bar{\sigma}^h = \inf\{ \, c^h x^h + (d + L^h_0 - L^h_1) y \, : \, (x^h, y) \in C^h \, \} \tag{55}$$

(where note that, unlike previously, y is a variable). By the assumption that C^h is the convex hull of S^h, $\bar{\sigma}^h = \sigma^h$: we can deal with (**??**) and (**??**) interchangeably. Similarly to before, it is also plain to see that $\sigma^h \, (= \bar{\sigma}^h) > -\infty$. Indeed, assume this was not the case for, say, $h = 1$; it would mean that there exists a direction $[\, v \, , \, w \,] \in rec(C^1)$ such that $c^1 v + dw < 0$. But y is bounded in S^1 (hence in C^1) by assumption, which means that any direction in the recession cone must have null w (the component corresponding to y). This would mean again that $[\, v \, , \, 0 \, , \, 0 \,] \in rec(C^1 \oplus C^2)$, and the scalar product with the objective is $c^1 v < 0$, which contradicts $\Pi > -\infty$.

We now need to introduce some notation to facilitate the subsequent discussion. First of all, each of the constraint functions f^1, f^2 and $f^3(x^1, x^2, y) = -y$—that of the single constraint $y \geq 0$—only depend on a subset of the variables: yet, in discussing the KKT conditions we have to consider them as functions of all the three sets of variables. We will however use shorthands, such as writing, say, $f^1(x^1, y)$ instead of $f^1(x^1, x^2, y)$ since x^2 are irrelevant. Similarly, the subdifferentials in principle have to be considered as subsets of $\mathbb{R}^{n_1+n_2+1}$; yet, for, say, $f^1(x^1, y)$ we will equivalently consider an element of $\partial f^1(x^1, y)$ as a vector $[v^1, w] \in \mathbb{R}^{n_1+1}$ or as the corresponding vector $[v^1, 0, w] \in \mathbb{R}^{n_1+n_2+1}$. The fundamental observation here is that the crucial property is of the form "$0 \in \partial f^1(x^1, y)$", which is equivalently satisfied in both cases since the components of the subgradients corresponding to variables that are not involved in the definition of the function are necessarily null.

Since the assumptions of the Lemma have to hold for each set separately (besides for $C^1 \oplus C^2$), each optimal solution $(x^h, y) \in C^h$ of (**??**) is a KKT point; that is, there exist Lagrangian multipliers $\mu^h \geq 0$ and $\lambda^h \geq 0$ such that

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \begin{bmatrix} c^h \\ d + L_0^h - L_1^h \end{bmatrix} + \mu^h \begin{bmatrix} 0 \\ -1 \end{bmatrix} + \sum_{i=1}^{m_h} \lambda_i^h \partial f_i^h(x^h, y) \tag{56a}$$

$$\mu^h y = 0 \tag{56b}$$

$$\lambda^h f^h(x^h, y) = 0 \tag{56c}$$

The crucial, albeit trivial to verify, observation is that $\bar{\sigma}^h = \sigma^h = L_0^h$ for both $h = 1, 2$: indeed, the objective value of (**??**) is L_0^h for both $y = 0$ and $y = 1$. Therefore, the problems (**??**) admit optimal solutions (them not being unbounded below) having *either* $y = 0$ *or* $y = 1$. Hence, we have at our disposal optimal primal solutions $(\bar{x}_y^h, y)$ and dual solutions $(\bar{\mu}_y^h, \bar{\lambda}_y^h)$—which necessarily satisfy (**??**)—for each $h = 1, 2$ and $y \in \{0, 1\}$. Let us immediately remark that, due to (**??**), $\bar{\mu}_1^h = 0$ (irrespectively of h); thus, we will denote $\bar{\mu}_0^h$ as $\bar{\mu}^h$. With the help of these solutions we can construct the sought-for primal and dual solutions, $(\hat{x}^1, \hat{x}^2, \hat{y})$ and $(\hat{\mu}, \hat{\lambda}^1, \hat{\lambda}^2)$ that satisfy (**??**). To do that, we have to distinguish two cases.

The first is $L = L_0 \leq L_1$, i.e., $y = 0$ is optimal in (**??**); here we take $(\hat{x}^1, \hat{x}^2, \hat{y}) = (\bar{x}_0^1, \bar{x}_0^2, 0)$ and $(\hat{\lambda}^1, \hat{\lambda}^2) = (\bar{\lambda}_0^1, \bar{\lambda}_0^2)$ (the value of $\hat{\mu}$ will be disclosed shortly). Now, (**??**)–(**??**) immediately follow from (**??**) for $h = 1, 2$, while (**??**) trivially holds since $y = 0$; thus, it remains to examine (**??**). For that, we combine (**??**) for $h = 1, 2$ to obtain

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \in \begin{bmatrix} c^1 \\ c^2 \\ (d + L_0^1 - L_1^1) + (d + L_0^2 - L_1^2) \end{bmatrix} + (\bar{\mu}^1 + \bar{\mu}^2) \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} + \sum_{i=1}^{m_1} \bar{\lambda}_{0,i}^1 \partial f_i^1(\bar{x}_0^1, 0) + \sum_{i=1}^{m_2} \bar{\lambda}_{0,i}^2 \partial f_i^2(\bar{x}_0^2, 0)$$

We have of course exploited the previously recalled property where the "$0 \in \partial \ldots$" condition still holds when extended to the irrelevant variables, and the fact that $0 \in P_1$ and $0 \in P_2$ implies $0 \in P_1 + P_2$ by definition of Minkowsky sum. Now, since $L_0 = L_0^1 + L_0^2$ and $L_1 = L_1^1 + L_1^2 - d$, one has

$$(d + L_0^1 - L_1^1) + (d + L_0^2 - L_1^2) = d - (L_1 - L_0) \ ,$$

which allows to rewrite the equation above as

$$\begin{bmatrix}0\\0\\0\end{bmatrix} \in \begin{bmatrix}c^1\\c^2\\d\end{bmatrix} + (L_1 - L_0 + \bar{\mu}^1 + \bar{\mu}^2)\begin{bmatrix}0\\0\\-1\end{bmatrix} + \sum_{i=1}^{m_1} \bar{\lambda}^1_{0,i}\partial f^1_i(\bar{x}^1_0, 0) + \sum_{i=1}^{m_2} \bar{\lambda}^2_{0,i}\partial f^2_i(\bar{x}^2_0, 0) \ .$$

Therefore, $\hat{\mu} = L_1 - L_0 + \bar{\mu}^1 + \bar{\mu}^2 \geq 0$ (since $\bar{\mu}^h \geq 0$ for $h = 1,2$ and $L_0 \leq L_1$) completes the definition of a dual solution satisfying (**??**). Hence,

$$\Delta = \Pi = c^1\hat{x}^1 + c^2\hat{x}^2 = L^1_0 + L^2_0 = L_0 = L$$

as desired.

For the case where, instead, $L = L_1 < L_0$, we must introduce the other problem(s)

$$\sigma = \inf\{\, (L - L_0)y \,:\, (x^1, y) \in S^1 \,\} = \inf\{\, (L - L_0)y \,:\, (x^1, y) \in C^1 \,\} \tag{57}$$

(the equivalence between the two being obvious). Since by definition $L - L_0 < 0$, each optimal solution of (**??**) must have $y = 1$, which in turn implies $\sigma = L - L_0 > -\infty$. Again, the hypotheses ensure that the (leftmost) problem in (**??**) has some optimal solution $(\tilde{x}^1, \tilde{y}) = (\tilde{x}^1, 1)$, which is a KKT point; therefore, it admits Lagrangian multipliers $\tilde{\lambda}$ ($\tilde{\mu} = 0$ for obvious reasons) such that

$$\begin{bmatrix}0\\0\end{bmatrix} \in \begin{bmatrix}0\\L - L_0\end{bmatrix} + \sum_{i=1}^{m_1} \tilde{\lambda}_i \partial f^1_i(\tilde{x}^1, 1) \tag{58a}$$

$$\tilde{\lambda} f^1(\tilde{x}^1, 1) = 0. \tag{58b}$$

Now, (**??**) is satisfied for any optimal solution $(\tilde{x}^1, 1)$ of (**??**). But the coefficients of the objective function corresponding to x^1 in (**??**) are null; this means that *any* feasible solution $(x^1, 1) \in C^1$ is optimal for (**??**). We are therefore free to choose $\tilde{x}^1 = \bar{x}^1_1$ (the optimal solution of (**??**) for $h = 1$ and $y = 1$). We want to show that $(\hat{\mu}, \hat{\lambda}^1, \hat{\lambda}^2) = (0, \bar{\lambda}^1_1 + \tilde{\lambda}, \bar{\lambda}^2_1)$ satisfies (**??**) with $(\hat{x}^1, \hat{x}^2, \hat{y}) = (\bar{x}^1_1, \bar{x}^2_1, 1)$. Indeed, as in the previous case, (**??**) immediately follows from (**??**) for $h = 2$ and $y = 1$, while (**??**) follows from the combination of (**??**) for $h = y = 1$ and (**??**) ((**??**) is trivial). We then have to combine (**??**) for $h = 1,2$ and $y = 1$ with (**??**) to yield

$$\begin{bmatrix}0\\0\\0\end{bmatrix} \in \begin{bmatrix}c^1\\c^2\\(d + L^1_0 - L^1_1) + (d + L^2_0 - L^2_1) + (L - L_0)\end{bmatrix} + \sum_{i=1}^{m_1} (\bar{\lambda}^1_{1,i} + \tilde{\lambda}_i)\partial f^1_i(\hat{x}^1, 1) + \sum_{i=1}^{m_2} \bar{\lambda}^2_{1,i}\partial f^2_i(\hat{x}^2, 1)$$

We now use again $L_0 = L^1_0 + L^2_0$ and $L_1 = L^1_1 + L^2_1 - d$, together with $L = L_1$, to get

$$(d + L^1_0 - L^1_1) + (d + L^2_0 - L^2_1) + (L - L_0) = d + (L_0 - L_1) + (L - L_0) = d$$

to get that, again, (**??**) holds, and therefore the whole of (**??**) does, for the chosen $(\hat{\mu}, \hat{\lambda}^1, \hat{\lambda}^2)$ and $(\hat{x}^1, \hat{x}^2, \hat{y})$. For the objective function value, let us remark that $\bar{\sigma}^h = L^h_0$ for $h = 1,2$ and (**??**) give

$$c\bar{x}^h_1 + (d + L^h_0 - L^h_1) = L^h_0 \implies c\hat{x}^h = c\bar{x}^h_1 = L^h_1 - d \ ,$$

whence

$$\Delta = \Pi = c^1\hat{x}^1 + c^2\hat{x}^2 + d = L_1^1 + L_1^2 - d = L_1 = L$$

in this case as well, finishing the proof. □

Appendix B: Model T

Here we report a full description of the model T summarized in Table 3 in **?**. This model corresponds to the *3-bin* that we have used as benchmark for our experiments. The formulation described in the following uses our notation but the constraints are numbered exactly as in referenced article. Note that power variables p_{it} are substituted by variables $p'_{it} = p_{it} - l_i x_{it}$. For any further details, refer to **?**.

$$x_{it} - x_{i,t-1} = v_{it} - w_{it} \qquad i \in I, t(2,n) \qquad (2)$$

$$x_{it} = 1 \qquad i \in I, t \in T(1, \tau_i^+ - \tau_i^0), \text{ if } \tau_i^0 > 0 \qquad (3_a)$$

$$x_{it} = 0 \qquad i \in I, t \in T(1, \tau_i^- + \tau_i^0), \text{ if } \tau_i^0 < 0 \qquad (3_v)$$

$$\textstyle\sum_{s \in T(t-\tau_i^+ + 1, t)} v_{is} \le x_{it} \qquad i \in I, t \in T(\tau_i^+, n) \qquad (4)$$

$$\textstyle\sum_{s \in T(t-\tau_i^- + 1, t)} w_{is} \le 1 - x_{it} \qquad i \in I, t \in T(\tau_i^-, n) \qquad (5)$$

$$p_{it} = p'_{it} + l_i x_{it} \qquad i \in I, t \in T \qquad (15)$$

$$p'_{it} \le (u_i - l_i)x_{it} - (u_i - \bar{l}_i)v_{it} - (u_i - \bar{u}_i)w_{i,t+1} \qquad i \in I, \tau_i^+ > 1, t \in T \qquad (20)$$

$$p'_{it} \le (u_i - l_i)x_{it} - (u_i - \bar{l}_i)v_{it} \qquad i \in I, \tau_i^+ = 1, \bar{l}_i = \bar{u}_i, t \in T \qquad (21_a)$$

$$p'_{it} \le (u_i - l_i)x_{it} - (u_i - \bar{u}_i)w_{i,t+1} \qquad i \in I, \tau_i^+ = 1, \bar{l}_i = \bar{u}_i, t \in T \qquad (21_b)$$

$$p'_{it} \le (u_i - l_i)x_{it} - (u_i - \bar{l}_i)v_{it} - [\bar{l}_i - \bar{u}_i]^+ w_{i,t+1} \qquad i \in I, \tau_i^+ > 1, \bar{l}_i \ne \bar{u}_i t \in T \qquad (23_a)$$

$$p'_{it} \le (u_i - l_i)x_{it} - [\bar{u}_i - \bar{l}_i]^+ v_{it} - (u_i - \bar{u}_i)w_{i,t+1} \qquad i \in I, \tau_i^+ > 1, \bar{l}_i \ne \bar{u}_i, t \in T \qquad (23_b)$$

$$p'_{it} \le (u_i - l_i)x_{it} - \textstyle\sum_{s=1}^{\min\{\tau^+ - 1, T^{RU}\}} (u_i - \bar{l}_i - s\Delta_i^+)v_{it-s} \qquad i \in I, \tau^+ - 2 < T^{RU}, t \in T \qquad (40)$$

$$p'_{it} - p'_{i,t-1} \le (\bar{l}_i - l_i - \Delta_i^+)v_{it} + \Delta_i^+ x_{it} \qquad i \in I, t \in (2,n) \qquad (35)$$

$$p'_{i,t-1} - p'_{it} \le (\bar{u}_i - l_i - \Delta_i^-)w_{it} + \Delta_i^- x_{i,t-1} \qquad i \in I, t \in (2,n) \qquad (36)$$

$$\textstyle\sum_{i \in I}(p'_{it} + l_i x_{it}) = d_t \qquad t \in T \qquad (65)$$

In (40), $T_i^{RU} = \lfloor \frac{u_i - \bar{u}_i}{\Delta_i^+} \rfloor$. Furthermore, for all $i \in I$, $t \in T$ we have

$$p'_{it} \le (u_i - l_i)x_{it} - \textstyle\sum_{s=1}^{\min\{\tau^+ - 2, T^{RU}\}} (u_i - \bar{l}_i - s\Delta_i^+)v_{it-s} - (u_i - \bar{u}_i)w_{i,t+1} \qquad (38)$$

The model also considers constraints (41). We have noticed that the definition of $K_i^{SU}(t)$ used for deifnining these constraints contains a typo. Consequently, we have corrected as follows.

Given $T_i^{RD} = \lfloor \frac{u_i - \bar{l}_i}{\Delta_i^-} \rfloor$, $K_i^{SD}(t) = \min\{T_i^{RD}, \tau_i^+ - 1, n - t - 1\}$ and $K_i^{SU}(t) = \min\{T_i^{RU}, \tau_i^+ - 2 - [K_i^{SD}(t)]^+, t - 1\}$, where $[\cdot]^+ := \max\{\cdot, 0\}$, for all $i \in I$, $t \in T$ such that $K_i^{SD}(t) > 0$ we have

$$p'_{it} \leq (u_i - l_i)x_{it} - \sum_{s=1}^{K_i^{SU}(t)} (u_i - \bar{l}_i - s\Delta_i^+)v_{it-s} - \sum_{s=1}^{K_i^{SD}(t)} (u_i - \bar{u}_i - s\Delta_i^-)w_{i,t+1+s} \tag{41}$$

Finally, because of the use of variables p'_{it} instead of p_{it}, the convex envelope (**??**) of the objective function (**??**) can be written as

$$h_i(p'_{it}, x_{it}) = \begin{cases} a_i \frac{p'^2_{it}}{x_{it}} + (2a_i l_i + b_i)p_{it} + (l_i^2 + l_i b)x_{it} & \text{if } x_{it} > 0 \\ 0 & \text{if } x_{it} = 0 \end{cases} \tag{59}$$

and consequently the objective function is

$$\min \; \sum_{i \in I} \sum_{t \in T} (\textbf{??}) + \sum_{i \in I} \sum_{t \in T} c_i x_{it} + \sum_{i \in I} (\textbf{??}) \tag{60}$$

Acknowledgments

The second author acknowledges the partial financial support by the European Union Horizon 2020 research and innovation programme under grant agreement No 773897 “plan4res”. The second and third authors acknowledge the partial financial support by the European Union Horizon 2020 Marie Skłodowska-Curie Actions under the grant agreement No 764759 “MINOA”. The first, the second, and the third authors acknowledge the partial financial support by the Italian Ministry of Education program MIUR-PRIN under the grant 2015B5F27W “Nonlinear and Combinatorial Aspects of Complex Networks”.

We are grateful to two anonymous referees and to the editors for their suggestions which helped us to considerably improve the quality of this paper.

Brief author biographies

Tiziano Bacci is a researcher in Optimization and Discrete Mathematics at the National Research Council of Italy. He graduated with honors in Management Engineering from the University of Rome Tor Vergata in 2012 and got his Ph.D. in Computer Science from the same University in 2018. He is an adjunct professor in Operations research at University of Rome Tor Vergata. His main research interest are the development of exact and heuristic algorithms for combinatorial optimization and Mixed-Integer Nonlinear optimization problems. The main applications are in energy, logistics and healthcare.

Antonio Frangioni graduated with honors in Computer Science from the University of Pisa, Italy, in 1992, got his Ph.D. in Computer Science from the same University in 1996, and spent all his career there, where he is now Full Professor since 2014. His main research interests are in models and algorithms for large-scale continuous and combinatorial optimization problems, using such techniques as decomposition algorithms, interior-point methods, reformulation techniques and network flow approaches, and applying them to diverse fields such as energy, telecommunications, logistics, and others. He is author and co-author of more than 70 journal articles, not counting several book chapters and conference presentations.

Claudio Gentile graduated with honors in Computer Science from the University of Pisa and from Scuola Normale Superiore of Pisa in 1995, and received his Phd in "Operations Research" at Sapienza University of Rome in 2000. He is research director at Institute for System Analysis and Computer Science "A. Ruberti" of the Italian National Research Council (IASI-CNR) since 2020, where he started his scientific career in 2001. His main research interests are in Combinatorial Optimization,Polyhedral Theory for Linear and Nonlinear Mixed-Integer Programming problems, Interior Point Methods, applications in Power Energy Production and Distribution, Logistics, Network Design, Staff Management, and Ship Scheduling. He is co-author of more than 50 scientific publications among journal papers, book chapters, and articles in conference proceedings.

Kostas Tavlaridis-Gyparakis got a B.Sc. in Financial & Management Engineering at the School of Bussiness of the University of the Aegean in 2012, a M.Sc. in Operations Research at the School of Mathematics of the University of Edinburgh in 2013, and from 2014 to 2018 he was Ph.D. student at the Doctorate in Computer Science of the University of Pisa, under the auspices of the PGMO Project "Consistent Dual Signals and Optimal Primal Solutions", where he achieved the Ph.D. title with the thesis "Decomposition Techniques for Large-Scale Energy Optimization Problems".