Electric Power Systems Research 160 (2018) 429–438

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

Effects of binary variables in mixed integer linear programming based unit commitment in large-scale electricity markets

Juan Alemany[a,*], Leszek Kasprzyk[b], Fernando Magnago[a,c]

[a] Power Systems Analysis Group, Río Cuarto National University, Argentina
[b] Institute of Electrical and Electronics Industry, Poznan University of Technology, Poland
[c] Nexant Inc., AZ, USA

ARTICLE INFO

Article history:
Received 16 August 2017
Received in revised form 22 March 2018
Accepted 24 March 2018

Keywords:
Binary variables relaxation
Branch and cut algorithm
Day-ahead electricity market clearing
Mixed integer linear programming
Unit commitment

ABSTRACT

Mixed integer linear programming is one of the main approaches used to solve unit commitment problems. Due to the computational complexity of unit commitment problems, several researches remark the benefits of using less binary variables or relaxing them for the branch-and-cut algorithm. However, integrality constraints relaxation seems to be case dependent because there are many instances where applying it may not improve the computational burden. In addition, there is a lack of extensive numerical experiments evaluating the effects of the relaxation of binary variables in mixed integer linear programming based unit commitment. Therefore, the primary purpose of this work is to analyze the effects of binary variables and compare different relaxations, supported by extensive computational experiments. To accomplish this objective, two power systems are used for the numerical tests: the IEEE118 test system and a very large scale real system. The results suggest that a direct link between the relaxation of binary variables and computational burden cannot be easily assured in the general case. Therefore, relaxing binary variables should not be used as a general rule-of-practice to improve computational burden, at least, until each particular model is tested under different load scenarios and formulations to quantify the final effects of binary variables on the specific UC implementation. The secondary aim of this work is to give some preliminary insight into the reasons that could be supporting the binary relaxation in some UC instances.

1. Introduction

The unit commitment (UC) calculation is extensively used in day-ahead power system operation planning and comprises the pre-dispatch of power generation units to satisfy the electricity demand and unit operation constraints. It is an exercise of large-scale, time-varying, non-convex, mixed-integer modeling and optimization. The average size of the UC problem for large electricity market applications is commonly conformed by hundred thousands of variables and constraints [1]. The size of this challenging problem is one of the primary motivations to develop different strategies to improve the algorithm's performance. Not only regarding computational times but also, to obtain more economy efficient global solutions.

Currently, the mathematical optimization technique known as Mixed-Integer Linear Programming (MILP, Branch-and-Cut and Heuristics based algorithms) is one of the main solvers applied to UC problems. MILP applied to the UC problem is not new [2–4]; however, it recently becomes popular due to new advances [5,6] that permit solving problems for the sizes of electricity markets and allow to develop more complex formulations. Different current day-ahead electricity market clearing applications support its applicability, for example [7–9]. MILP model flexibility and accuracy, as well as state-of-the-art algorithms, have evolved over the years to become robust and effectively enough to fulfill the electricity market necessities, overpassing the performance of Lagrange Relaxation (LR) based applications, and in some cases replacing them, as the most popular algorithms used in the past.

Some works [10,11] have compared the computational behavior between the MILP and LR, UC based algorithms. The general conclusion extracted is that the LR algorithm has a linear computational behavior and the MILP algorithm has an exponential behavior with the increase of the UC size. Based on this exponential behavior, the number of binary variables in a MILP based UC is often used as an indicator of the computational difficulty in several works [12–14]. This intuitive assumption, probably not based

* corresponding author.
E-mail address: jalemany@ing.unrc.edu.ar (J. Alemany).

https://doi.org/10.1016/j.epsr.2018.03.019
0378-7796/

on exhaustive numerical observation, gives reasonable room to the suggestion of using fewer or relaxing some of the binary variables to obtain the consequent improvement in computational performance.

In the general case, the Branch-and-Cut (B&C) based algorithms discard large sections of the tree of potential solutions from further examination. This is the main difference between B&C algorithms and exhaustive enumeration. The B&C algorithms detect infeasible solutions or out-of-bounds solutions and sequentially prune the branches that form the tree of potential solutions. In this process, the solution space is gradually reduced until reaching the convergence criteria. This criterion is usually a time limit or a gap distance between the lower and upper bounds. These bounds are defined by the last best integer solution and its relaxation (i.e., the linear program solution). The solution obtained after convergence is always a feasible solution. This fact might be the most remarkable feature of B&C based algorithms, compared to LR based ones.

Nevertheless, modern B&C solvers (CPLEX,GUROBI) implement pre-solver heuristics mainly taking advantage of fixing variables, tightening bounds of variables, guessing preliminary values of variables, eliminating redundant rows and columns of the constraints matrix, changing the sense of constraints, reducing symmetry. The main purpose of the pre-solver heuristics is to find fast a feasible solution of the problem (integer solution) before starting the regular B&C process. The heuristics of the pre-solver eventually facilitate the problem resolution. However, it can be experimentally observed that relaxing binary variables to achieve better computational performance, produces the risk of losing the benefits of the pre-solver strength. Therefore, this binary-relaxation rule-of-thumb method should not be considered as a general rule-of-practice, because it may not take the advantages of the pre-solver. In consequence, the number of binary variables might become a poor indicator of the computational difficulty when solving a MILP based realistic UC [15].

The commercial success of solvers like CPLEX or GUROBI, is mainly due to their robustness and fast performance. This solver evolution has produced that many market operators become to develop their own market clearing tools; therefore, nowadays they focus their efforts in modeling and not in algorithms. Consequently, it is really important to evaluate different model performances under different circumstances like the number of binary variables used, always thinking on real life applications.

Although there are many current works [16–19] focused on model tightness and "good" constraints, the focus of our work is in the number of binary variables (and in the relaxation of the integrality constraints that bind them) utilized in the UC model.

In this regard, the primary objective of this work is to evaluate through extensive numerical simulations the effects of relaxing the binary variables when solving a MILP based UC for a real large-scale electricity market. The secondary aim of this work is to give some preliminary insight on the reasons that could be supporting the binary relaxation in some UC instances.

The work is organized as follow: Section 3 formulates the UC model used for the purpose of this work. Section 5 presents the numerical results and the discussion about them; Finally, Section 7 resumes the main conclusions of this work.

2. Nomenclature

Sets

T, G	Time and unit sets.
t, g	Hour and unit index set.
B	Piece-wise block set.
b	Power block index set.
n, N	Index and set for the multi-steps start-up costs.

Variables

Cp, Cs	Unit's production and start-up costs.
δ_{bgt}	Unit's piece-wise block variable.
p_{gt}	Unit's active power variable.
r_{gt}	Unit's reserve variable.
u_{gt}	Unit's state binary variable.
j_{bgt}	Unit's power block binary variable.
s_{gt}, h_{gt}	Unit's start-up/shut-down binary variables.
z^*	Objective value for feasible solution.
$z_{(u,\ldots)}$	Simulation instances for different binary variables.

Parameters

D_t	Hourly system demand.
R	Reserve requirement.
τ	Unit's number of steps for start-up costs.
$K_{g\tau}$	Unit's start-up cost for step τ.
K_g	Unit's constant start-up cost.
c_g	Unit's generation fix cost.
F_{bg}	Unit's power block slope.
Tr_{bg}	Unit's min-max power block limits.
UT_g, DT_g	Unit's min on/off service times.
T_g^{on}, T_g^{off}	Unit's initial on/off hours of service.
$\underline{P}_g$, $\bar{P}_g$	Unit's min-max power limits.
RU_g, RD_g	Unit's up-down ramp limits.

3. Unit commitment model

For real market applications, the traditional UC problem [12,10,20] is formulated as a cost minimization function subject to system, units and a representative set of some critical network constraints. In general, the minimization function considers the generation costs, including production costs, start-up costs, and no-load costs.

Currently, many works present different UC formulations with improved computational efficiency [22–24]. In [22] the authors propose a projection technique to simplify some constraints of the UC, allowing the solvers to exploit it. The effectiveness of the methodology is demonstrated through a set of randomly generated UC cases, with sizes ranging from 10 to 200 units over a 24 hours scheduling horizon. In [23] the authors present a MILP model for an accurate representation of the main technical and operating characteristics of thermal generation units on a day-ahead market, incorporating non-convex production costs, time-dependent start-up costs, and inter-temporal constraints. The effectiveness of the model is demonstrated with an UC case conformed by 15 thermal units and 24 h scheduling horizon. In [24], the authors present two improved formulations for reducing the approximation errors produced by the traditional interpolation function for costs. The proposed optimal linear approximation and the optimal piecewise linear approximation are more precise and reduce the computational times. The effectiveness of the formulation is demonstrated with a set of 36 thermal units and up to 96 scheduling periods.

However, none of these publications present results for large-scale real power market applications [1]. The model used in this work is based on traditional models proposed in [12,10,20] which have been used effectively in the electricity industry. Network constraints were not included in the model in order to make the explanation simpler without losing generality since the main objective of this work is not altered by the inclusion of network constraints. The reason for this, depends on the mathematical nature of the network constraints; they can be linear or nonlinear relationships among continuous variables. These constraints do not affect the branching process of a MIP algorithm because this process exclusively depends on the binary variables.

Additionally, in market practices can be considered that the most important feature of "network constraints" is related to the addition of security constraints [21]. In general, for large scale real power systems applications, it is non-viable to directly consider the large set of likely contingencies; therefore, decomposition techniques are commonly utilized because they permit to separate the

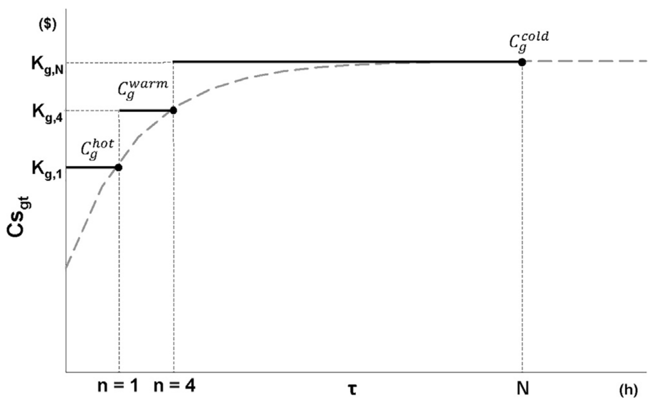

Fig. 1. Multi-step start-up costs.

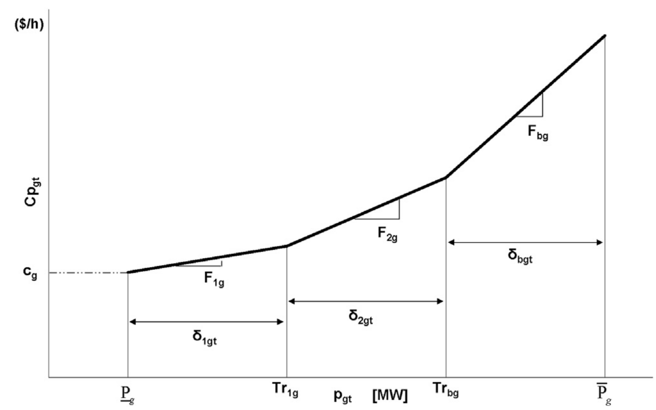

Fig. 2. Convex production costs.

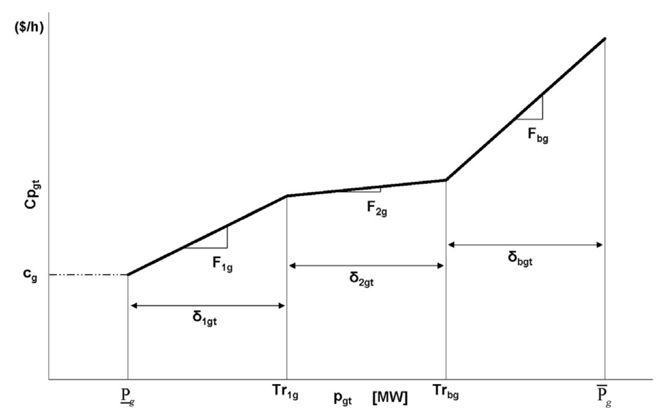

Fig. 3. Non-convex production costs.

UC resolution from the security checking phase. In this regard, the main conclusions drawn in the paper will not be altered by the exclusion of network constraints.

The detailed mathematical formulation for the objective function as well as for the constraints is reproduced next. The model describes the binary variables that will be included or relaxed during the simulation tests.

$$\min z = \sum_{t=1}^{T} \sum_{g=1}^{G} (\mathrm{Cp}_{gt} + \mathrm{Cs}_{gt}) \tag{1}$$

$$\mathrm{Cp}_{gt} = u_{gt}\ c_g + \sum_{b=1}^{B} \mathrm{F}_{bg}\ \delta_{bgt} \quad \forall gt \tag{2}$$

$$\mathrm{Cs}_{gt} \geq \mathrm{K}_{g\tau}\left(u_{gt} - \sum_{n=1}^{\tau} u_{g,t-n}\right) \quad \forall gt\tau \tag{3}$$

$$\mathrm{Cs}_{gt} \geq 0 \quad \forall gt \tag{4}$$

$$\mathrm{Cs}_{gt} = s_{gt}\mathrm{K}_g \quad \forall gt \tag{5}$$

where Eq. (1) is the objective function; Eq. (2) represents the production costs; Eqs. (3) and (4) represent the stepwise start-up costs; and Eq. (5) represents the one-step start-up costs. In this work, are considered convex and non-convex production costs depending on the units. For the non-convex piecewise formulation an extra binary variable is needed j_{bgt}. Also, two star-up cost formulations are used depending on the units. A stepwise formulation which is able to model hot, warm and cold costs (multi-step formulation) and a single-step formulation for these cycling costs. It is important to note that the multi-step formulation does not need an extra binary variable; on the contrary, the single-step formulation does it. Fig. 1 illustrates a multi-step start-up costs representation.

Additional constraints need to be added in the MILP formulation of the production costs:

$$p_{gt} = u_{gt}\ \underline{\mathrm{P}}_g + \sum_{b=1}^{B} \delta_{bgt} \quad \forall gt \tag{6}$$

$$p_{gt} \geq 0 \quad \forall gt \tag{7}$$

$$(\mathrm{Tr}_{1g} - \underline{\mathrm{P}}_g) j_{1gt} \leq \delta_{1gt} \leq (\mathrm{Tr}_{1g} - \underline{\mathrm{P}}_g) u_{gt} \quad \forall gt \tag{8}$$

$$(\mathrm{Tr}_{bg} - \mathrm{Tr}_{b-1,g}) j_{bgt} \leq \delta_{bgt} \quad \forall bgt \tag{9}$$

$$\delta_{bgt} \leq (\mathrm{Tr}_{bg} - \mathrm{Tr}_{b-1,g}) j_{b-1,gt} \quad \forall bgt \tag{10}$$

$$\delta_{\mathrm{B}gt} \geq 0 \quad \forall gt \tag{11}$$

$$\delta_{\mathrm{B}gt} \leq (\bar{\mathrm{P}}_g - \mathrm{Tr}_{\mathrm{B}-1,g}) j_{\mathrm{B}-1,gt} \quad \forall gt \tag{12}$$

These equations, Eqs. (6)–(12), are the piece-wise linear formulation of production costs represented by Eq. (2). Note that binary variable j_{bgt} is used for the non-convex cases only. Fig. 2 shows a representation of convex production costs and Fig. 3 illustrates non-convex ones.

System constraints includes energy balance constraints and reserve constraints. Eq. (13) represents the total energy balance. Eqs. (14) and (15) represent the primary reserve requirements that must be met by the assigned units.

$$\sum_{g=1}^{G} p_{gt} = \mathrm{D}_t \quad \forall t \tag{13}$$

$$\sum_{g=1}^{G} r_{gt} \geq \mathrm{D}_t\ (1 + \mathrm{R}) \quad \forall t \tag{14}$$

$$p_{gt} + r_{gt} \leq u_{gt}\bar{\mathrm{P}}_g \quad \forall gt \tag{15}$$

Then, unit constraints which include the binary variables are included into the set of constraints:

$$\sum_{i=t-\mathrm{UT}_g+1}^{t} s_{gi} \leq u_{gt} \quad \forall g \forall t \in [\mathrm{UT}_g + 1,\ \mathrm{T}] \tag{16}$$

$$\sum_{i=t-\mathrm{DT}_g+1}^{t} h_{gi} \leq 1 - u_{gt} \quad \forall g \forall t \in [\mathrm{DT}_g + 1,\ \mathrm{T}] \tag{17}$$

$$\sum_{i=0}^{i \leq \mathrm{T}_g^{on}} 1 - u_{gi} = 0 \quad \forall g \forall t = 0 \tag{18}$$

Table 1
Convex and one-step costs: methodologies comparison.

$\mathbb{B}$ vars.	B&C				B&B				Evol.			
	u	*u, s*	*u, h*	*u, s, h*	*u*	*u, s*	*u, h*	*u, s, h*	*u*	*u, s*	*u, h*	*u, s, h*
Peak = 1625 MW z^* = 489,515												
Nodes	2110	5528	276	445	7335	3510	3000	2126	1	1	1	1
Time	3.89	9.96	3.07	3.07	35.77	23.12	21.67	17.93	1.28	1.50	2.17	1.51
Gap	0.0	0.0	0.0	0.0	0.07	0.27	0.28	0.38	2.12	2.12	2.12	2.11
Peak = 2166 MW z^* = 658,573												
Nodes	34	23	27	39	159	565	210	684	1	1	1	1
Time	2.64	2.70	2.80	2.90	2.67	5.41	3.84	6.19	1.26	1.75	1.50	2.20
Gap	0.0	0.0	0.0	0.0	0.52	0.27	0.41	0.23	1.14	1.14	1.14	1.14
Peak = 2708 MW z^* = 838,459												
Nodes	3179	3271	3018	3655	450	3480	3481	2664	1	2	1	2
Time	14.6	18.6	19.9	34.8	5.51	20.17	21.92	19.92	3.74	5.79	1.70	6.52
Gap	0.0	0.0	0.0	0.0	0.26	0.01	0.01	0.01	0.51	0.49	0.51	0.49
Peak = 3250 MW z^* = 1,028,388												
Nodes	20,832	40,914	434,927	1,920,722	1350	160	500	230	1	2	1	1
Time	23.1	30.9	276.6	1000*	7.55	3.32	6.40	5.71	2.67	5.79	2.79	2.25
Gap	0.0	0.0	0.0	–	0.01	0.15	0.01	0.14	0.23	0.23	0.23	0.23
Peak = 3791 MW z^* = 1,223,101												
Nodes	0	0	0	0	0	0	0	0	1	1	1	1
Time	1.09	1.13	1.19	1.19	0.28	0.38	0.36	0.38	0.50	0.61	0.59	0.62
Gap	0.0	0.0	0.0	0.0	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11
Peak = 4333 MW z^* = 1,428,566												
Nodes	831	1043	5244	1417	50	170	0	0	1	1	1	1
Time	2.08	3.22	5.45	3.24	1.87	3.74	0.61	0.55	1.22	1.34	1.14	1.08
Gap	0.0	0.0	0.0	0.0	0.16	0.09	0.21	0.21	0.21	0.20	0.21	0.21
Peak = 4874 MW z^* = 1,641,164												
Nodes	2124	2810	1018	1018	0	4304	60	3060	1	1	1	4
Time	11.6	12.6	8.43	9.64	0.38	21.59	3.70	17.99	3.04	3.59	1.76	20.66
Gap	0.0	0.0	0.0	0.0	0.38	0.04	0.31	0.08	0.38	0.38	0.38	0.38
Peak = 5416 MW z^* = 1,862,776												
Nodes	27,320	64,624	33,364	30,739	100	194,353	86,000	5000	1	4	3	3
Time	367.3	576.9	674.8	553.9	4.31	1000*	432.40	33.35	11.22	69.64	19.97	17.86
Gap	0.0	0.0	0.0	0.0	0.62	0.18	0.23	0.41	0.68	0.65	0.65	0.65

$$\sum_{i=0}^{i\le T_g^{off}} u_{gi} = 0 \quad \forall g \forall t = 0 \tag{19}$$

$$u_{gt}\underline{P}_g \le p_{gt} \le u_{gt}\bar{P}_g \quad \forall gt \tag{20}$$

$$p_{gt} - p_{g,t-1} \le RU_g + s_{gt}\underline{P}_g \quad \forall g \forall t > 1 \tag{21}$$

$$p_{g,t-1} - p_{gt} \le RD_g + h_{gt}\underline{P}_g \quad \forall g \forall t > 1 \tag{22}$$

$$s_{gt} - h_{gt} = u_{gt} - u_{g,t-1} \quad \forall gt \tag{23}$$

$$s_{gt} + h_{gt} \le 1 \quad \forall gt \tag{24}$$

where Eqs. (16)–(19) represent min up-down times; Eq. (20) represents the generator operational limits; Eqs. (21) and (22) are the ramp limits; and Eqs. (23) and (24) represent the logic among binary variables. Within this formulation, can be seen that there are several variables that are binary: the status variable u_{gt}; the start-up variable s_{gt}; the shut-down variable h_{gt}; and the non-convex costs variable j_{bgt}. It is important to note that, with exception of u_{gt} variables, the other binary variables can be relaxed to be continuous between 0 and 1. This is because, once u_{gt} variables have been set to 0/1 states by the algorithm, the set of constraints (23) and (24) force the rest of relaxed binary variables to obtain only binary values.

The formulation described by Eqs. (1)–(24) represents the traditional UC model. This model considers the main technical and operational characteristics of: the generation resources like production and start-up costs, ramp and service time limits; and the energy system coupling constraints like energy balance and reserve requirements. Therefore, it offers a good balance among accuracy, flexibility and robustness to solve the general UC problem.

4. Methodology settings

The simulations consider the model represented by Eqs. (1)–(24). First, only the status variable u_{gt} is considered as binary, all other binary variables (s_{gt}, h_{gt}, and j_{bgt}) are relaxed. In this context, relaxation means that they are set as continuous variables in the region {0,1}, however, Eqs. (23) and (24) force relaxed variables to take {0,1} values despite its continue nature. These variables are included in the binary set, one at a time to perform different simulation cases. Both convex and non-convex cost curves are considered, the main difference is the addition of variable j_{bgt} for the non-convex case. Also, one-step and step-wise start-up costs are considered. Shutdown costs are in general not considered for practical large systems. No analysis was made, regarding the number of power blocks to piece-wise the cost functions.

The model is implemented in GAMS-CPLEX [25] using default options for branch and cut methodology (B&C). For pure branch and bound methodology (B&B) *mipsearch* option is set to `1` and cuts are deactivated. For Evolutionary methodology (Evol.) *mipsearch* option is set to `1` and CPLEX Evolutionary algorithm [26] is activated. Two different test systems are used; the IEEE-118 system [27] that comprises 54 thermal units and a large scale real system with 2321 thermal units. Eight different load profiles are used for both systems, the peak value for these loads are set from 1625 MW to 5416 MW for the first system and from 19,153 MW to 268,149 MW for the second system. The total simulation time is set to 24 h, and the simulation step to 1 h, representing a typical day-ahead study time frame. No impact on the main objective of the article is observed using ramp rate constraints; therefore, they were discarded from the simulations.

Table 2
Non convex and step-wise costs: B&C and B&B comparison.

$\mathbb{B}$ vars.	B&C								B&B							
	u	u,s	u,h	u,j	u,s h	u,s j	u,h j	u,s h,j	u	u,s	u,h	u,j	u,s h	u,s j	u,h j	u,s h,j
Peak = 1625 MW	z^* = 93,740															
Nodes	0	1890	1410	520	420	0	610	0	5000	2590	21,000	3000	15,000	14,000	31,000	30,000
Time	6.90	228.2	171.3	114.5	50.9	8.31	125.2	7.35	62.46	32.58	187.08	47.97	153.01	203.43	390.83	397.35
Gap	1.71	0.24	0.75	1.08	1.40	1.79	0.89	1.80	4.91	5.61	3.60	5.26	3.76	4.55	3.73	3.63
Peak = 2166 MW	z^* = 128,496															
Nodes	290	490	380	300	3840	2544	640	720	72,500	107,700	99,000	80,400	117,100	21,000	18,800	85,400
Time	26.1	49.8	28.7	41.3	337.7	190.9	75.4	64.1	952.82	916.76	927.64	917.60	906.35	226.11	197.84	912.70
Gap	1.20	1.13	1.11	1.02	0.30	0.42	0.92	1.17	1.83	1.10	1.39	1.43	1.06	2.66	2.73	2.10
Peak = 2708 MW	z^* = 169,988															
Nodes	4510	11,732	10,710	11,132	4510	4616	1506	4590	95,921	97,000	107,220	76,900	120,600	86,800	88,900	88,500
Time	168.1	344.3	219.8	673.1	180.3	165.4	111.9	139.1	1000⋆	919.92	1000⋆	930.30	959.97	906.87	1000⋆	953.68
Gap	0.44	0.29	0.26	0.68	0.32	0.48	0.50	0.37	2.17	2.39	2.07	2.22	1.54	1.93	2.44	2.20
Peak = 3250 MW	z^* = 217,641															
Nodes	6508	1016	1100	9104	360	1000	1512	7005	96,500	94,000	80,000	17,000	98,400	26,000	5000	22,000
Time	708.3	102.2	74.9	923.5	39.9	164.6	188.6	1000⋆	911.28	911.22	794.56	196.36	915.99	267.71	61.48	257.04
Gap	0.55	0.77	0.84	0.49	0.82	0.57	0.49	0.51	2.67	2.47	2.48	3.44	2.53	3.24	4.11	3.34
Peak = 3791 MW	z^* = 275,116															
Nodes	4080	2070	3699	5227	9152	5070	6175	9575	110,800	112,900	114,200	89,700	101,317	78,800	83,900	60,000
Time	448.4	228.7	331.1	531.5	797.8	418.9	619.0	914.3	907.08	929.42	916.68	905.15	1000⋆	932.47	913.00	662.97
Gap	0.61	0.50	0.36	0.50	0.33	0.41	0.37	0.37	4.03	3.97	3.90	4.06	4.26	4.28	3.99	4.07
Peak = 4333 MW	z^* = 332,946															
Nodes	1000	6479	180	11,431	270	520	562	950	99,500	99,800	93,100	80,200	78,600	28,000	33,000	23,000
Time	98.9	311.1	17.3	644.7	19.8	50.3	55.8	94.6	906.59	911.67	971.03	908.32	917.30	412.84	374.75	249.79
Gap	0.50	0.30	0.63	0.25	0.63	0.62	0.59	0.44	2.92	2.43	3.16	2.84	3.45	3.93	3.82	3.50
Peak = 4874 MW	z^* = 412,901															
Nodes	4101	10,059	6958	10,437	8479	9975	3475	10,450	106,100	157,400	109,000	118,300	130,000	56,000	62,000	56,090
Time	32.7	49.3	38.2	69.9	41.2	50.6	24.1	63.4	527.47	912.59	680.93	906.21	858.96	410.07	432.14	415.94
Gap	0.16	0.15	0.14	0.14	0.15	0.12	0.13	0.11	0.44	0.18	0.48	0.33	0.35	0.61	0.56	0.51
Peak = 5416 MW	z^* = 557,775															
Nodes	597	150	100	320	360	615	490	342	5641	8100	6040	1000	9000	3100	1100	7469
Time	8.96	3.59	3.34	4.54	6.60	8.00	4.59	4.46	39.30	53.77	46.33	13.81	58.58	31.50	16.94	58.39
Gap	0.02	0.00	0.03	0.01	0.00	0.02	0.00	0.02	0.04	0.04	0.05	0.69	0.04	0.38	0.81	0.26

Performance improvement is defined as the computational time decrease with respect to the reference case z_u.

5. Numerical results

The numerical results are presented in tables, the columns of these tables correspond to the variables set as binary, variables not shown in these columns are set as continuous variables. The rows indicate the different load profiles used during the simulations, which are characterized by their maximum peak in MW. The objective value z^* is the variable set as the stop criteria for all simulation cases. For the convex cases, the zero duality gap is also set as the final stop criteria. Reported times and gaps are in seconds and percentage respectively. Symbol (⋆) identifies the cases where the maximum simulation time was reached and the objective cost was not reached before this time.

5.1. IEEE 118 bus system

Results carried out using this system are summarized in Tables 1–3.

Table 1 presents the results obtained with the IEEE-118 considering all units with convex production costs and one-step start-up costs. Based on the results of Table 1, relaxation seems to be more convenient with B&C methodology for this particular case. For example, $z_{u,s}$ cases were not a good option compared to the base cases z_u. Although for low peaks, there were some $z_{u,s,h}$ cases with improved performances. In particular, every time $z_{u,h}$ cases had better performances than z_u cases then $z_{u,s,h}$ cases have had better performances than z_u cases. With Evolutionary methodology, again, relaxation is found to be more convenient for this case. Relaxation cases overcame 6 times over 2 binary augmented cases. For example, $z_{u,s}$ cases would be never recommended with this methodology for this particular case. Alternatively, results with B&B methodology show that relaxation is convenient in some cases but not always. In this case, the simulation patterns seem to be more diverse, having 5 relaxation cases with improved performances compared to 3 binary augmented cases with improved performances. However, avoiding $z_{u,s}$ cases seem to be convenient again. The same as before, when $z_{u,h}$ cases overcame z_u cases then $z_{u,s,h}$ cases were even superior than z_u cases.

Tables 2 and 3 presents the results on the IEEE-118 considering non-convex production costs and step-wise start-up costs. Table 2 shows the comparison results between B&C methodology with B&B methodology, while Table 3 illustrates the comparison results with respect to the Evolutionary methodology. Based on the results of Tables 2 and 3, they seem to be more promissory for binary augmented cases compared to the convex case. Indeed, this is confirmed by the general results obtained with the B&B and Evolutionary methodologies. For example, full binary cases $z_{u,s,h,j}$ surpasses 6 times over 2 base cases z_u. In addition, under these particular circumstances, many binary augmented cases could be found as better implementations, with improved computational performances, compared to base cases z_u. Instead, with the B&C methodology, the results obtained were very much diverse and while it is possible to find augmented cases which are substantially better than z_u cases, there is not a clear advantage toward binary augmented cases. For example, $z_{u,j}$ cases were never a good imple-

Table 3
Non convex and step-wise costs: evolutionary.

$\mathbb{B}$ vars.	*u*	*u, s*	*u, h*	*u, j*	*u, s* *h*	*u, s* *j*	*u, h* *j*	*u, s* *h, j*
Peak = *1625 MW* z^* = 93,740								
Nodes	1	2	2	2	2	1	3	1
Time	6.77	8.80	8.28	10.67	12.51	4.49	16.99	5.29
Gap	6.60	6.60	6.60	6.60	6.60	4.11	6.60	2.45
Peak = *2166 MW* z^* = 128,496								
Nodes	3	3	2	1	3	3	4	2
Time	82.35	46.83	47.52	8.06	87.10	26.97	173.91	10.37
Gap	5.35	5.21	5.34	2.36	5.21	1.67	5.34	2.47
Peak = *2708 MW* z^* = 169,988								
Nodes	3	7	5	8	3	2	8	4
Time	123.29	335.17	199.90	464.56	94.88	16.24	301.08	81.85
Gap	4.89	4.93	4.93	2.47	4.89	1.85	2.04	2.55
Peak = *3250 MW* z^* = 217,641								
Nodes	1	2	2	1	3	2	2	2
Time	56.68	25.26	24.10	60.36	52.70	24.06	15.05	19.10
Gap	5.09	5.07	5.09	5.09	5.07	5.06	5.06	2.29
Peak = *3791 MW* z^* = 275,116								
Nodes	1	2	1	1	2	2	2	2
Time	44.04	19.89	13.53	55.12	12.22	22.03	10.55	20.56
Gap	5.54	5.54	5.53	5.54	5.54	5.54	5.54	5.54
Peak = *4333 MW* z^* = 332,946								
Nodes	1	2	2	1	3	2	2	2
Time	34.98	35.44	26.88	49.16	50.72	30.25	15.55	40.30
Gap	5.05	5.07	5.07	5.05	5.06	1.21	1.43	5.07
Peak = *4874 MW* z^* = 412,901								
Nodes	1	2	2	3	2	4	2	2
Time	14.98	28.52	21.47	46.07	17.18	84.55	10.30	16.91
Gap	2.71	2.70	2.54	2.51	2.54	0.80	0.89	2.71
Peak = *5416 MW* z^* = 557,775								
Nodes	1	1	2	2	2	4	3	1
Time	5.52	3.71	7.64	13.10	11.61	39.75	25.88	4.32
Gap	1.75	1.75	1.75	1.69	1.75	0.54	0.54	1.75

Table 4
Real system.

$\mathbb{B}$ vars.	*u*	*u, s*	*u, h*	*u, s, h*
Peak = *19,153 MW* z^* = 1,683,702				
Nodes	0	0	0	0
Time	39.7	24.2	52.2	43.7
Peak = *38,307 MW* z^* = 3,421,306				
Nodes	76	19	17	70
Time	39.8	39.2	36.6	88.9
Peak = *76,614 MW* z^* = 10,320,353				
Nodes	493	343	443	101
Time	85.2	64.2	114.7	96.9
Peak = *114,921 MW* z^* = 20,006,906				
Nodes	296	79	171	59
Time	99.0	55.8	68.6	55.0
Peak = *153,228 MW* z^* = 31,509,249				
Nodes	190	245	321	195
Time	81.6	57.5	30.1	59.7
Peak = *191,535 MW* z^* = 46,383,624				
Nodes	2147	1967	1863	979
Time	109.0	86.3	100.5	79.8
Peak = *229,842 MW* z^* = 68,147,951				
Nodes	60	1784	1933	2367
Time	115.1	97.2	56.0	103.2
Peak = *268,149 MW* z^* = 99,393,089				
Nodes	78	40	50	30
Time	70.2	61.0	54.7	78.1

Table 5
Results CPLEX 9.

	90% of the peak demand							
	MILP-3Bin	MILP-3Bin Anti-Sym	MILP-1Bin	MILP-1Bin Anti-Sym	MILP-3Bin	MILP-3Bin Anti-Sym	MILP-1Bin	MILP-1Bin Anti-Sym
110 units								
Objective [$]	6,200,278	6,203,516	6,205,742	6,204,557	5,447,453	5,447,029	5,448,158	5,446,253
Time [s]	21.9	16.0	12.0	16.6	99.1	24.8	900	900
Gap [%]	0.23	029	0.48	0.49	0.46	0.45	0.64	0.55
120 units								
Objective [$]	6,759,804	6,765,928	6,768,137	6,762,374	5,941,263	5,943,982	5,943,536	5,944,525
Time [s]	47.5	15.2	20.2	14.2	46.5	27.2	900	900
Gap [%]	0.17	0.26	0.46	0.41	0.44	0.48	0.67	0.64
130 units								
Objective [$]	7,326,277	7,326,705	7,336,930	7,333,958	6,437,934	6,440,345	6,442,194	6,436,351
Time [s]	11.4	15.8	356.9	19.5	21.0	50.2	900	900
Gap [%]	0.21	0.21	0.49	0.49	0.47	0.49	0.73	0.54
140 units								
Objective [$]	7,892,100	7,886,835	7,894,441	7,893,165	6,935,335	6,936,035	6,934,438	6,931,502
Time [s]	13.1	20.6	62.7	24.5	23.7	28.7	900	900
Gap [%]	0.28	0.19	0.47	0.48	0.49	0.49	0.68	0.60
150 units								
Objective [$]	8,452,143	8,457,615	8,453,242	8,458,277	7,430,680	7,431,086	7,429,591	7,431,133
Time [s]	17.1	19.2	21.4	35.4	109.8	120.9	900	900
Gap [%]	0.22	0.28	0.40	0.49	0.49	0.49	0.69	0.70
160 units								
Objective [$]	9,018,570	9,017,290	9,022,892	9,023,994	7,923,416	7,940,242	7,931,575	7,927,146
Time [s]	19.5	28.0	29.7	38.6	60.1	64.4	900	900
Gap [%]	0.25	0.24	0.44	0.48	0.46	0.47	0.78	0.68
170 units								
Objective [$]	9,581,612	9,577,360	9,589,246	9,586,129	8,418,008	8,419,357	8,422,943	8,419,426
Time [s]	22.6	27.1	50.1	15.6	139.5	79.2	900	900
Gap [%]	0.24	0.19	0.49	0.48	0.46	0.49	0.72	0.66
180 units								
Objective [$]	10,139,587	10,141,474	10,153,210	10,151,116	8,911,511	8,915,485	8,917,061	8,913,397
Time [s]	26.8	25.4	71.2	18.2	59.8	45.8	900	900
Gap [%]	0.18	0.19	0.48	0.49	0.45	0.48	0.72	0.67
190 units								
Objective [$]	10,701,543	10,705,403	10,713,569	10,705,480	9,410,883	9,407,178	9,411,990	9,409,143
Time [s]	24.3	33.6	206.9	30.0	94.9	63.8	900	900
Gap [%]	0.19	0.22	0.47	0.43	0.49	0.44	0.72	0.66
200 units								
Objective [$]	11,269,906	11,274,924	11,277,725	11,274,267	9,898,606	9,902,813	9,903,291	9,909,886
Time [s]	30.5	38.5	92.8	17.9	77.9	80.2	900	900
Gap [%]	0.23	0.27	0.47	0.47	0.41	0.46	0.67	0.74

mentation option, although in general, $z_{u,h,j}$ cases seem to be better than $z_{u,s,h,j}$ cases when comparing against z_u cases. Therefore, under any circumstance can be assured that z_u cases are better than any binary augmented cases.

5.2. Large scale real system

Additional simulations were carried out using the large scale real system conformed by 2321 thermal units. CPLEX default options were utilized in these simulations. Table 4 shows the results on the large system considering convex production costs and one-step start-up costs. All the gaps reached 0% within the 1000 s (default running time).

Results presented in Table 4 illustrate that including start-up variables as binary, $z_{u,s}$ cases, improves the performance for all load scenarios, and the average improvements are about 35% of computational time. On the contrary, the addition of binary shut-down variables, $z_{u,h}$ cases, seems not to be such a good option as before because 2 cases have a bad performance compared to the base cases z_u, and the 3 effectively improved cases coincide with the binary start-up augmented cases $z_{u,s}$. In addition, considering all variables as binary, $z_{u,s,h}$ cases, a better performance was obtained in 3 of the simulated cases, other 4 cases are invariant, and only 1 case is considerably bad affected. On the whole, the results for this system suggest that binary relaxation could be surpassed by thorough addition of binary variables.

6. Discussion of the effects of binary variables

One of the purposes of Section 5 was to construct a stronger insight about the effects of binary relaxation on the UC problem, supported by extensive numerical experimentation with a real large-scale electricity market. From the results, it is not correct to assure that reducing the number of binary variables – by relaxing the integrality constraints over some integer variables in a UC problem – may not always result in faster computation times. Therefore, relaxation of binary variables should not be used as a general rule-of-practice to improve computational burden for short-term thermal UC problems. In average, the use of models with modern solvers and with more binary variables was more beneficial for the computational burden.

Table 6
Results CPLEX 12.

	90% of the peak demand			
	MILP-3Bin	MILP-1Bin	MILP-3Bin	MILP-1Bin
100 units				
Objective [$]	5,643,900	5,638,807	4,954,295	4,951,590
Time [s]	3.2	3.4	5.1	8.3
Gap [%]	0.38	0.38	0.25	0.49
150 units				
Objective [$]	8,453,137	8,453,117	7,427,948	7,423,967
Time [s]	4.8	4.2	8.4	15.2
Gap [%]	0.19	0.36	0.28	0.45
200 units				
Objective [$]	11,278,763	11,275,493	9,896,566	9,894,559
Time [s]	7.9	6.7	14.2	21.4
Gap [%]	0.25	0.38	0.15	0.45
250 units				
Objective [$]	14,098,508	14,085,190	12,368,348	12,371,782
Time [s]	12.1	14.3	17.4	29.2
Gap [%]	0.27	0.27	0.12	0.49
300 units				
Objective [$]	16,923,668	16,897,022	14,836,093	14,851,858
Time [s]	12.3	21.2	26.5	39.7
Gap [%]	0.28	0.23	0.08	0.49
350 units				
Objective [$]	19,744,224	19,726,963	17,308,411	17,320,631
Time [s]	18.0	24.5	29.1	43.0
Gap [%]	0.27	0.37	0.08	0.46
400 units				
Objective [$]	22,560,792	22,563,942	19,778,328	19,782,352
Time [s]	23.5	37.0	34.3	54.4
Gap [%]	0.26	0.45	0.06	0.41
450 units				
Objective [$]	25,380,013	25,363,156	22,252,137	22,266,032
Time [s]	32.1	25.4	55.7	62.4
Gap [%]	0.25	0.39	0.08	0.47
500 units				
Objective [$]	28,190,111	28,166,796	24,720,626	24,741,854
Time [s]	40.2	25.0	53.8	80.7
Gap [%]	0.21	0.38	0.06	0.49

The contribution of this section is to discuss some of the reasons why the computational results presented in this work, have differences with respect to previous ones.

6.1. Previous work

In [13] the authors analyze the dominance of different constraints of the traditional UC model. Basically, the authors discuss the dominance of two sets of typical constraints in the UC, the minimum up/down times of the units and the logic relationships among the binary variables u_{gt}, s_{gt}, h_{gt}. For these sets of constraints, it is worth to note that the formulation used in our work and in [13] are identical. The only difference is the inclusion of Eq. (24) in our work. Dismissing constraints like Eq. (24) could lead, in some of the simulation periods, to set s_{gt} and h_{gt} variables to one, which is physically impossible. The authors of work [13], suggest using Eqs. (16), (17), (23) with binary variables u_{gt}, and relaxing integrality for variables s_{gt}, h_{gt}, i.e., defining $0 \le s_{gt}$, $h_{gt} \le 1$.

In [14] the authors perform a comparison between the MILP-UC version (a model consisting only on u_{gt} binaries) with a MILP-3 version (a model consisting of three binaries u_{gt}, s_{gt}, h_{gt}). However, while the detailed formulation of the MILP-UC version is presented there, the MILP-3 is only referenced to other articles. In our case, we used the same model (Eqs. (1)–(24)) for the numerical comparisons and it is quite similar to the one described in [14]. The only difference is the simpler ramp formulation (Eqs. (21) and (22)) in our work; however, ramp constraints are not used neither for the simulation of this section nor for the simulation in [14]. In work [14] the authors suggest using less binary variables on the UC model to obtain computational improvements.

Regarding the suggestion proposed in [13], in Section 5, we presented extensive numerical experimentation to inquire about the usefulness of this suggestion, at least, with modern solvers as CPLEX 12. Regarding the suggestion in [14], in the next section we present further results to expand the insight about its effects.

6.2. Further results

We analyze here the results obtained applying the model formulation of the work [14] using the same simulation scenarios with CPLEX 9. The same parameter settings have been taken into account; namely, relative gap 0.5%, reserve requirement of 10%, start-up costs modeled by a 15 stair-wise linear function, production cost linearized through a piecewise linear approximation with 4 segments, and load demand accordingly multiplied to fulfill all the load scenarios. Ramping constraints have not been taken into account because the work [14] does not consider them in the simulations. Shutdown costs have been set to zero.

Table 5 presents the results for CPLEX 9. The particularity with these scenarios (110 units to 200 units) is that they are all symmetrical [1]. The simulation scenarios in [14] are created multiplying the basic data base (10 units) as many times as necessary to generate instances of up to 100 units. This process generates a highly symmetrical problem which causes computational issues on any MILP based UC application [1]. Under these circumstances, it is expected that reducing the number of binaries will reduce the computational burden; mainly because, symmetry issues arise with the impossibility to differentiate among binary variables. The results presented in [14] confirm this statement at least until the 100 units case. Nevertheless, the results presented in Table 5 show a different behavior from the case with 130 units and larger systems; where, the MILP-3Bin formulation produce better performances than the MILP-1Bin formulation. Using antisymmetry measures [1] seems to reduce drastically, the computational burden of the worst cases for the MILP-1Bin model version.

Please observe that Table 5 contains a new peak load instance of 90%. The addition of the 90% Peak instance is not casual. As it was mentioned in [1], in some cases it is likely to have bigger symmetry issues when the load demand allows the solution space problem to contain more feasible options; i.e., more combinations to be explored by the algorithm. Under this load scenario the results are totally different than the expected from predictions given in reference [14]. The MILP-1Bin formulation fails to provide a feasible solution below the gap setting and the algorithm stops by exceeding the time limit setting. Antisymmetry measures seems to be useless to provide a convergence improvement; although in general, all the final gaps obtained with antisymmetry cuts are better than the plain MILP-1Bin formulation. Regarding the MILP-3Bin formulation, 50% of the cases are improved with antisymmetry cuts, 40% of the cases are invariant and only 1case is heavily deteriorated.

Once again, CPLEX 12 – default settings – have been used with extended simulation scenarios up to 500 units. Table 6 presents the results. The new version of CPLEX, applies pre-solver heuristics and antisymmetry cuts that produce, in general, better results with formulations that contain more binaries to model the MILP based UC. Here, we only present an extract of simulation results; however, this performance has been observed in numerous simulation cases with different power systems.

7. Conclusion

In this work, we were interested in questioning whether binary variables should be relaxed in the MILP based UC problem. While not guaranteed, it is a common literature recommendation to treat a variable as continuous (instead of binary) and this generally may lead to improved computational performance. However, as it was demonstrated in this work, there are exceptions to this case and we wished to examine such a situation for the UC problem. We showed using different simulation scenarios, that results are disappointingly inconclusive.

First of all, predictability in MILP based UC seems to be a thankless task. Second, depending on the needs of an electricity market, they can include convex or non-convex formulations for representing production costs of the generation units. The same is valid for the start-up costs, with variants of single step formulations or multi-step formulations. Therefore, a general purpose application for the clearing of the day-ahead electricity market should include all the modeling possibilities that could meet the customer's needs.

Based on this work, it seems impossible to generalize a unique behavior in relation with binary variables and the different existing models in the literature. Instead, it is recommendable to test any user model under different circumstances, like for example different load scenarios, before being able to obtain any valuable disclosure possibly based on theoretical assumptions. Under the MILP based UC problem scenario, extensive numerical experimentation applied to realistic power systems, seems to be a better option to generate valuable conclusions.

Specifically, our results demonstrated that:

- Contrary to the expected, there existed many cases when a bigger set of binary variables improved the performance of our UC (model) resolution, especially with the new solver versions.
- Regrettably, a direct link between relaxed binary variables and computational burden could not be assured. At least for the current results, the relation seemed to be of unpredictable behavior.

Acknowledgment

This work was funded by: CONICET, Argentina; Secretaría de Ciencia y Técnica, Universidad Nacional de Río Cuarto, Argentina; and Alexander von Humboldt Stiftung, Deutschland. The authors would like to acknowledge the contributions made by these organizations.

References

[1] J. Alemany, P. Komarnicki, J. Lin, F. Magnago, Exploiting symmetry in unit commitment solutions for a large-scale electricity market, Electr. Power Syst. Res. 140 (Nov) (2016) 363–377.
[2] L. Garver, Power generation scheduling by integer programming – development of theory, Trans. Am. Inst. Electr. Eng. Power Appar. Syst. 81 (April) (1962) 730–734.
[3] J. Muckstadt, R. Wilson, An application of mixed-integer programming duality to scheduling thermal generating systems, IEEE Trans. Power Appar. Syst. 87 (December) (1968) 1968–1978.
[4] A. Cohen, S. Wan, A method for solving the fuel constrained unit commitment problem, IEEE Trans. Power Syst. 2 (August) (1987) 608–614.
[5] R. Bixby, The Latest Advances in MIP Solvers, Spring School on Combinatorial Optimization in Logistics, Universite de Montreal, 2010, May.
[6] R. Bixby, M. Fenelon, Z. Gu, E. Rothberg, R. Wunderling, in: Powell, Scholtes (Eds.), "MIP: Theory and practice – closing the gap", System Modeling and Optimization: Methods, Theory, and Applications, Kluwer Academic Publishers, 2000.
[7] Y. Chen, A. Casto, F. Wang, Q. Wang, X. Wang, J. Wan, Improving large scale day-ahead security constrained unit commitment performance, IEEE Trans. Power Syst. 31 (November) (2016) 4732–4743.
[8] P. Ristanovic, J. Waight, CAISO market redesign and technology update, IEEE Power Engineering Society General Meeting, vol. 3 (2005) 2636–2637.
[9] D. Streiffert, R. Philbrick, A. Ott, A mixed integer programming solution for market clearing and reliability analysis, IEEE Power Engineering Society General Meeting, vol. 3 (2005) 2724–2731.
[10] T. Li, M. Shahidehpour, Price-based unit commitment: a case of Lagrangian relaxation versus mixed integer programming, IEEE Trans. Power Syst. 20 (November (4)) (2005) 2015–2025.
[11] X. Guan, Q. Zhai, A. Papalexopoulos, Optimization based methods for unit commitment: Lagrangian relaxation versus general mixed integer programming, IEEE Power Engineering Society General Meeting, vol. 2 (2003) 1100.
[12] A. Viana, J. Pedroso, A new MILP-based approach for unit commitment in power production planning, Int. J. Electr. Power Energy Syst. 44 (January) (2013) 997–1005.
[13] K. Hedman, R. O'Neill, S. Oren, Analyzing valid inequalities of the generation unit commitment problem, in: IEEE PES, Power Systems Conference and Exposition, March, 2009.
[14] M. Carrion, J. Arroyo, A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem, IEEE Trans. Power Syst. 21 (August) (2006) 1371–1378.
[15] H. Williams, Model Building in Mathematical Programming, 5th Ed., John Wiley & Sons, Ltd., England, 2008.
[16] R. Fuentes-Loyola, V. Quintana, Medium-term hydro-thermal coordination by semi-definite programming, IEEE Trans. Power Syst. 18 (2003) 1515–1522.
[17] X. Bai, H. Wei, Semi-definite programming-based method for security-constrained unit commitment with operational and optimal power flow constraints, IET Gener. Transm. Distrib. 3 (February (2)) (2009) 182–197.
[18] J. Ostrowski, M. Anjos, A. Vannelli, Tight mixed integer linear programming formulations for the unit commitment problem, IEEE Trans. Power Syst. 27 (1) (2012) 39–46.
[19] G. Morales, J. Latorre, A. Ramos, Tight and compact milp formulation for the thermal unit commitment problem, IEEE Trans. Power Syst. 28 (4) (2013) 4897–4908.
[20] T. Dillon, K. Edwin, H. Kochs, R. Taud, Integer programming approach to the problem of optimal unit commitment with probabilistic reserve determination, IEEE Trans. Power Appar. Syst. 6 (November) (1978) 2154–2166.
[21] B. Stott, O. Alsac, Optimal Power Flow – Basic Requirements for Real-Life Problems and Their Solutions, 2012, July https://pdfs.semanticscholar.org/673e/416fa38c04611e90b12cbdb1a74ca08367db.pdf.

[22] L. Yang, J. Jian, Y. Wang, Z. Dong, Projected mixed integer programming formulations for unit commitment problem, Int. J. Electr. Power Energy Syst. 68 (June) (2015) 195–202.
[23] S. Bisanovic, M. Hajro, M. Dlakic, Unit commitment problem in deregulated environment, Int. J. Electr. Power Energy Syst. 42 (November) (2012) 150–157.
[24] Q. Zhai, X. Guan, J. Yang, Fast unit commitment based on optimal linear approximation to nonlinear fuel cost: error analysis and applications, Electr. Power Syst. Res. 79 (November) (2009) 1604–1613.
[25] General, Algebraic Modeling System, 2015 http://www.gams.com.
[26] E. Rothberg, An evolutionary algorithm for polishing mixed integer programming solutions, INFORMS J. Comput. 19 (July) (2007) 534–541.
[27] Illinois Institute of Technology (2013), http://motor.ece.iit.edu/data/JEAS_IEEE118.