

UNIVERSIDADE ESTADUAL DE CAMPINAS
SISTEMA DE BIBLIOTECAS DA UNICAMP
REPOSITÓRIO DA PRODUÇÃO CIENTÍFICA E INTELECTUAL DA UNICAMP

Versão do arquivo anexado / Version of attached file:

Versão do Editor / Published Version

Mais informações no site da editora / Further information on publisher's website:

<https://www.cambridge.org/engage/miir/article-details/6111a05018911d3251dd4022>

DOI: <https://doi.org/10.33774/miir-2022-ztpbq>

Direitos autorais / Publisher's copyright statement:

©2022 by Cambridge University Press (CUP). All rights reserved.

DIRETORIA DE TRATAMENTO DA INFORMAÇÃO

Cidade Universitária Zeferino Vaz Barão Geraldo

CEP 13083-970 – Campinas SP

Fone: (19) 3521-6493

<http://www.repositorio.unicamp.br>

Analysis of infeasible unit-commitment solutions arising in energy optimization

Leonardo Delarmelina Secchin³, Guilherme Ramalho¹, Claudia Sagastizábal², Paulo J. S. Silva⁴, Kenny Vinente⁵

¹ *Câmara de Comercialização de Energia Elétrica (CCEE) Brazil*

^{2,4} *Universidade Estadual Campinas (UNICAMP) Brazil*

³ *Universidade Federal do Espírito Santo (UFES) Brazil* ⁵ *Universidade Federal do Amazonas*

(Communicated to MIIR on 4 January 2022)

Study Group: 6th Brazilian Study Group with Industry, 22-26 March 2021, São Carlos <http://www.cemeai.icmc.usp.br/6WSMPI/>

Communicated by: Francisco Louzada Neto and José Alberto Cuminato

Industrial Partner: Chamber of Electric Energy Commercialization (CCEE)

Presenter: Guilherme Ramalho

Team Members: Dr. Roberto Andreani, Dr. Claudia Sagastizábal, Dr. Paulo J. S. Silva, Dr. Sandra A. Santos (Universidade Estadual de Campinas – UNICAMP, Brazil), Mr. Arthur Henrique Sousa Cruz (Universidade Federal de Lavras – UFLA, Brazil), Dr. Juan Pablo Luna (Universidade Federal do Rio de Janeiro – UFRJ, Brazil), Dr. Guilherme Ramalho (Câmara de Comercialização de Energia Elétrica – CCEE, Brazil), Dr. Leonardo D. Secchin (Universidade Federal do Espírito Santo – UFES, Brazil), Dr. Edilaine Martins Soler (UNESP, Brazil). Prof. Kenny Vinente (Universidade Federal do Amazonas – UFAM, Brazil),

Industrial Sector: Energy/Utilities

Tools: Mathematical optimization, CPLEX

Key Words: Security-constrained unit-commitment, feasible dispatch, mathematical optimization

MSC2020 Codes: 49M29, 90C90

Summary

The day-ahead problem of finding optimal dispatch and prices for the Brazilian power system is modeled as a mixed-integer problem, with nonconvexities related to fixed costs and minimal generation requirements for some thermal power plants. The computational tool DESSEM is currently run by the independent system operator, to define the dispatch for the next day in the whole country. DESSEM also computes marginal costs of operation that CCEE, the trading chamber, uses to determine the hourly prices for energy commercialization. The respective models sometimes produces an infeasible output. This work analyzes theoretically those infeasibilities, and proposes a prioritization to progressively resolve the constraint violation, in a manner that is sound from the practical point of view. Pros and cons of different mathematical formulations are analyzed. Special attention is put on robustness of the model, when the optimality requirements for the unit-commitment problem vary.

1 Introduction and context

As a follow-up of the industrial problem dealt with in 2018 and 2019, on “Day-ahead pricing mechanisms for hydro-thermal power systems”, in partnership with CCEE, CEPEL and RADIX as industrial partners, in 2021 the study group focused on the dynamics of hourly prices when industrial consumers are demand responsive under the coordination of Juan Pablo Luna, Claudia Sagastizábal, and Paulo J. S. Silva.

Demand response is currently being tested by the Brazilian independent system operator, ONS, and by the trading chamber, CCEE, [1]. The program considers reductions of consumptions of some registered clients as an alternative to dispatching thermal power plants out of the merit order. The rationale is that when consumers with a flexible demand adapt their load to the energy prices the reliability of the transmission system increases and end consumers pay lower prices.

The day-ahead problem of finding optimal dispatch and prices for the Brazilian system is modeled as a mixed-integer linear programming problem, with nonconvexities related to fixed costs and minimal generation requirements for some thermal power plants [coin]. The computational tool DESSEM, developed by CEPEL [5], is currently run by ONS to define the dispatch for the next day for the whole country. Having the optimal dispatch, DESSEM computes marginal costs of operation that CCEE uses as a basis to determine the hourly prices for energy commercialization. To fit the different needs of the each involved party, the tool is developed in two variants. The model used by the operator, ONS, details transmission lines and generating units. By contrast, the tool employed by the commercialization chamber, CCEE, only models large lines that interchange energy within macro areas. There is a disparity between the system seen by the model defining the dispatch and the model defining the price. From a mathematical optimization viewpoint, those differences materialize as DESSEM providing an output that is not feasible.

The work, carried on using an academic version of DESSEM, analyzes theoretically those infeasibilities, and proposes a prioritization to progressively resolve the constraint violation, in a manner that is sound from the practical point of view. Pros and cons of different mathematical formulations are analysed. Special attention is put on robustness of the model, when the optimality requirements for the unit-commitment problem vary.

2 Mathematical Formulation of the UC problem solved by DESSEM

Selecting the time steps $t = 1, \dots, T$ along the planning horizon, a set $\mathcal{S}$ that comprises the thermal and hydro-power units, and the related parameters, the preliminary model is given by

$$\begin{aligned}
 \min_{(p_i, u_i)_{i \in \mathcal{S}}} \quad & \sum_{i \in \mathcal{S}} \text{GCost}_i(p_i, u_i) + \alpha_{\text{FCF}} \\
 \text{s.t.} \quad & \sum_{i \in \mathcal{S}} p_i^t = D^t, & t = 1, \dots, T \\
 & (p_i, u_i) \in \mathcal{Q}_i, & i \in \mathcal{S} \\
 & \pi_{\text{FCF}_0^k} + \sum_{i=1}^{\text{NH}} \pi_{\text{FCF}, v_i^k} V_i^T \leq \alpha_{\text{FCF}}, \quad k = 1, \dots, \text{N}_{\text{FCF}}
 \end{aligned} \tag{2.1}$$

in which α_{FCF} and the related constraints represent the future cost (of the water) function and the i -th unit generation cost is

$$\text{GCost}_i(p_i, u_i) := \sum_{t=1}^T (C_i(p_i^t) + F_i^+[u_i^t - u_i^{t-1}]_+ + F_i^-[u_i^{t-1} - u_i^t]_+). \tag{2.2}$$

The operational constraints are described by the set

$$\mathcal{Q}_i := \{(p_i, u_i) \in \mathcal{P}_i : u_i \in \{0, 1\}^T\}, \quad i \in \mathcal{S} \tag{2.3}$$

and, for $i \in \mathcal{S}$, we have that

$$(p_i, u_i) \in \mathcal{P}_i \quad \text{contains} \quad \begin{cases} p_{i, \min} u_i^t \leq p_i^t \leq p_{i, \max} u_i^t, & t = 1, \dots, T \\ |p_i^t - p_i^{t-1}| \leq \Delta p_i, & t = 1, \dots, T. \end{cases} \tag{2.4}$$

Additionally, for $t = 1, \dots, T$, water balance and reservoir bound constraints are stated as

$$V_h^t - V_h^{t-1} + \eta_h p_h^t = I^t, \tag{2.5}$$

and

$$V_h^{\min \text{ safe}} \leq V_h^t \leq V_h^{\max}, \tag{2.6}$$

with $V_h^{\min \text{ safe}} := \max(V_h^{\min}, 0.20V_h^{\max})$.

3 Additional constraints related to the Brazilian electrical system

Many typical infeasibilities are a result of additional constraints that were not captured in the toy model (2.1) above. In order to deal with more realistic examples we needed to add more detail to the model such as:

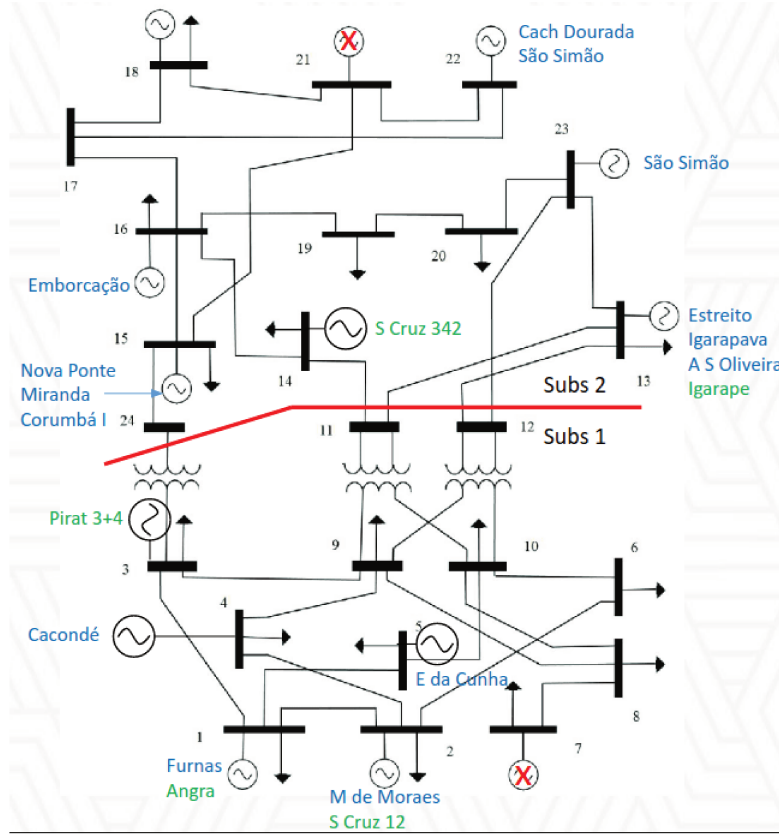


Figure 1. Graphical representation of the toy instance.

- Multiple bars;
- Power transmission lines between bars and related flux constraints;
- Possibly more than one subsystem;
- Hydro-power units in a cascade and associated constraints (spillage, water balance, etc.).

Figure 1 represents the test case used in the numerical section in the end of this report.

4 Identifying infeasibilities

Due to the enormous complexity of real instances, Problem (2.1) may be infeasible. In this case, the operator of the electrical system of Brazil (ONS) reports an infeasible (p^*, u^*) to the price decision maker (CCEE). The infeasibility measures are aggregated by adding slacks variables in the constraints. For simplicity, we consider that all constraints can be violated.

We want to eliminate the infeasibilities by changing bounds on the violated constraints as little as possible. So, we want to choose which constraints should be modified to minimize the infeasibility. This choice also must take into account the deterioration of the original objective value of (2.1) computed by the operator. Furthermore, we fix the

commitment $\mathbf{u}^*$. There are two reasons for this: (i) we want to maintain the operator's decision about the plants in operation at each time step, since it involves operational costs; (ii) we have a limited time budget to take a decision about infeasibilities. In the last case, fixing $\mathbf{u} = \mathbf{u}^*$ leads to solve pure continuous linear programs instead of MIPs.

To simplify the exposition, we see (2.1) as a general LP model with only inequality constraints, and fixing binary variables $\mathbf{u}$ at $\mathbf{u}^*$:

$$\begin{aligned} \min_{\mathbf{p}} \quad & \mathbf{c}^T \mathbf{p} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{p} \leq \mathbf{b}. \end{aligned} \tag{4.1}$$

Suppose that this problem is infeasible. We then add slack variables and penalize them, resulting in the problem

$$\begin{aligned} \min_{\mathbf{p}, \mathbf{s}} \quad & \mathbf{c}^T \mathbf{p} + M\mathbf{1}^T \mathbf{s} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{p} - \mathbf{s} \leq \mathbf{b} \\ & \mathbf{s} \geq \mathbf{0}, \end{aligned} \tag{4.2}$$

where $M > 0$ is the parameter of penalization and $\mathbf{1}$ is the vector of 1's. Evidently, there are other ways to penalize slacks; however, as the real instances have a huge size, the sum of slacks is reasonable since it results in a linear model that can be handled efficiently by standard solvers. Also, we assume that the penalization M is the same for all slacks. To simplify, **we suppose that fixing $\mathbf{u}$ leads to a feasible problem** (4.2). Notice that we consider being possible to eliminate all infeasibilities in a manner that is reasonable in the real world, i.e, achieving feasibility with relatively small changes in $\mathbf{b}$. This is an important issue, but for simplicity we do assume such a compatibility (otherwise, we probably should select some u_i 's to be relaxed).

We want to choose what constraints $\mathbf{a}_i^T \mathbf{p} - s_i \leq b_i$ have the major impact on the reduction of global infeasibility, measured by the penalization term $M\mathbf{1}^T \mathbf{s}$, when we change b_i .

5 Proposed strategy

We present a heuristic approach to reduce violated constraints that makes a systematic prioritization of constraints that are to be relaxed when infeasibility is detected by DESSEM. Alternative methodologies, that were considered but not implemented, are described in the Appendix A.

A natural approach to choose which constraints of (4.2) should be adjusted is to select first those with the largest associated Lagrange multipliers. Unfortunately, we show next that the Lagrange multipliers from a solution of (4.2) can not be of any help.

The optimality (KKT) conditions of (4.2) are

$$\mathbf{c} + \mathbf{A}^T \mathbf{y} = \mathbf{0}, \tag{5.1a}$$

$$M\mathbf{1} - \mathbf{y} - \mathbf{z} = \mathbf{0}, \tag{5.1b}$$

$$\mathbf{A}\mathbf{p} - \mathbf{s} \leq \mathbf{b}, \tag{5.1c}$$

$$\mathbf{s} \geq \mathbf{0}, \quad \mathbf{y} \geq \mathbf{0}, \quad \mathbf{z} \geq \mathbf{0}, \tag{5.1d}$$

$$\mathbf{y}^T (\mathbf{A}\mathbf{p} - \mathbf{s} - \mathbf{b}) = 0 \tag{5.1e}$$

$$z_i s_i = 0, \forall i, \quad (5.1f)$$

where $\mathbf{y}$ and $\mathbf{z}$ are the Lagrange multipliers vectors associated with constraints $\mathbf{A}\mathbf{p} - \mathbf{s} - \mathbf{b} \leq \mathbf{0}$ and $-\mathbf{s} \leq \mathbf{0}$, respectively. Multiplying the i -th row in (5.1b) by s_i and using (5.1f), we obtain, for each i ,

$$s_i y_i = M s_i.$$

Thus, all multipliers y_i associated with constraints $\mathbf{a}_i^T \mathbf{p} - s_i \leq b_i$ equals to M if the correspondent constraint is violated ($s_i > 0$). So, Lagrange multipliers do not reveal what constraints we have to adjust. It is worth be mentioned that if we use a different penalization parameter M_i for each slack variable, the associated y_i equals to M_i whenever $s_i > 0$. Again, Lagrange multipliers can not be used in this case either because they do not carry relevant information; they only follow the pre-defined empirically chosen M_i 's.

One conclusion of the above discussion is that the choice of the violated constraint(s) that should be adjusted are driven by M_i and $\mathbf{s}$ themselves. One possible strategy is to choose the constraints with largest slacks. Another more sophisticated approach is to choose a violated constraint that leads the maximal improvement on a criteria function after re-optimizing (4.2) (or some other related problem). We address these two ideas. In any case, changing the right-hand side of constraints in (4.2) may induce a solution $\mathbf{p}$ very distant from the original $\mathbf{p}^*$. We then must take into account this deterioration when adjusting violated constraints.

In general, we can formulate the algorithmic framework as Algorithm 1.

In the implementation we make use of a penalization M_i for each slack s_i . This can represent a specialist decision to order the constraints by its importance for the real world problem. Also, M_i may encompass a scale factor in the following sense: if the constraint $\mathbf{a}_i^T \mathbf{x} \leq b_i$ has a large data (for example, $|b_i| \gg 1$), we can scale its slack aiming to reduce the impact of a small relative change on b_i by using $\tilde{M}_i = M_i / \max\{1, |b_i|\}$.

6 Preliminary numerical tests

We tested the heuristic to reduce the number of violated constraints described in Algorithm 1 using an alternative implementation of the unit commitment problem developed by Kenny Vinente. The main advantage, with respect to using directly DESSEM, is that it allow us to capture and manipulate the final MIP model, as required in the algorithm.

In our tests we used the Null displacement function F described in (F1), based on the production cost. As for the expected reduction we opted for the (V2) alternative. It selects at each step the slack that can be eliminated and results in minimal deterioration of the production cost.

We started from the base instance described in Figure 1 and added extra constraints to make it infeasible:

- The thermal-power units were turned off for $t \geq 12$;
- Limited spillage for the hydro-power units;
- A subset of the power lines have limit flux capacity.

This case resulted in 19 positive slacks, that correspond to infeasibility. The heuristics of Algorithm 1 was able to eliminate 7 of those slacks with a minimal impact on the

Algorithm 1 Heuristics framework

Required: The initial $(\mathbf{p}^*, \mathbf{u}^*)$ solution of the (infeasible) problem (2.1), the vector of slacks $\mathbf{s}^*$, the *displacement function* $F(\mathbf{p})$ and its target value $\bar{F}$, and reduction on the global infeasibility $V(\mathbf{s})$. Define the set of indexes of violated constraints as $S = \{i \mid s_i^* > 0\}$.

Parameter: $k_{\max} > 0$.

Initialization: Fix $\mathbf{u} = \mathbf{u}^*$, initialize $k \leftarrow 0$, $\mathbf{s}^0 = \mathbf{s}^*$, the working set $\underline{S} \leftarrow \emptyset$ and the set of acceptable solutions $B \leftarrow \emptyset$. Define the problem

$$\min_{\mathbf{p}, \mathbf{s}} F(\mathbf{p}) + \sum_i M_i s_i \quad \text{s.t.} \quad \mathbf{A}\mathbf{p} - \mathbf{s} \leq \mathbf{b}, \quad \mathbf{s} \geq \mathbf{0}, \quad s_i = 0, \quad i \in \underline{S} \quad (\text{P})$$

($s_i = 0$ means that s_i can be eliminated from the model).

- (1) If $k > k_{\max}$ or $\underline{S} = S$, go to Step 7.
- (2) Compute the expected reduction $\mathcal{V}_i$ on infeasibility for each slack $s_i^k > 0$, $i \in S \setminus \underline{S}$. If all $\mathcal{V}_i$'s are equal to $-\infty$, go to the Step 7.
- (3) Add to $\underline{S}$ the index associated with the largest $\mathcal{V}_i$'s computed in the previous step (ties are broken randomly). Update model (P).
- (4) Solve (P) if it has not yet been solved. If (P) is feasible, take its solution $(\mathbf{p}^k, \mathbf{s}^k)$ and continue. Otherwise, go to Step 7.
- (5) If $F(\mathbf{p}^k) \leq \bar{F}$, update $B \leftarrow B \cup \{(\mathbf{p}^k, \mathbf{s}^k)\}$.
- (6) Take $k \leftarrow k + 1$ and go to Step 1.
- (7) If $B \neq \emptyset$, stop and return the best candidate found, that is, the one with the smallest number of slacks. In this case, adjust the right-hand side $\mathbf{b}$ accordingly. Otherwise, stop declaring failure.

overall cost of problem (4.2) of only 0.03% that comes only from increasing the violation of the remaining infeasibilities, increasing the respective slack values. The production cost remained the same, while the slack component of the objective increase by only 5.40%.

Another test case was obtained allowing slacks only on the constraints associated to transmission lines. In this case, the final number of infeasibilities increased to 22. The heuristics succeeded on identifying 6 slacks that could be eliminated. Once more this was achieved keeping the production cost intact and increasing the slack component by only 6.21%. This results in an overall increase of 0.03% of the original cost of problem (4.2).

Conclusions

To conclude, it is worth mentioning a few remarks that can be useful when implementing the proposal in the real time process.

Depending on the specialist experience, it may be mandatory to maintain the feasibility of the constraints that were already feasible in the ‘‘MIP phase’’. In this case, slacks should

not be added to these constraints. Otherwise, a previously feasible constraint may become infeasible after solving the penalized problem.

Regarding the parameter $k_{\max}$ in Algorithm 1, we observe that

- Its choice should consider the total computational time budget available.
- Whenever the model (P) is modified, we do not need to re-optimize it from the scratch. This must be reflected in the implementation. Once such a strategy is adopted, we expect the computational time to be reduced, and consequently, $k_{\max}$ may be increased.

In Algorithm 1, we have to specify the displacement function F and how the expected reductions $\mathcal{V}_i$ are computed. For F , we list some possible choices:

- (F1) **F based on the cost of production.** We measure the displacement of $\mathbf{p}$ in relation to $\mathbf{p}^*$ *indirectly* by the original function of the problem (2.1), that is, $F(\mathbf{p})$ is exactly the objective function $F_1(\mathbf{p})$ of (2.1) with $\mathbf{u} = \mathbf{u}^*$. In this case the target value can be $\bar{F} = \tau F_1(\mathbf{p}^*)$ where $\tau > 1$ is a parameter. This approach aims to maintain the total *energy production cost* unaltered as much as possible;
- (F2) **F based on the plan of production.** Here, we want to minimize the displacement of $\mathbf{p}$ *directly*. To maintain the resulting model linear we choose

$$F(\mathbf{p}) = F_2(\mathbf{p}) = \sum_i |p_i - p_i^*| = \sum_i pd_i$$

by means of additional continuous variables pd_i and constraints $pd_i \geq p_i - p_i^*$, $pd_i \geq -p_i + p_i^*$, $pd_i \geq 0$ for all i . Here, $\bar{F}$ represents a tolerance in the distance between $\mathbf{p}$ and $\mathbf{p}^*$;

- (F3) **Null F .** The simplest choice is $F \equiv \mathbf{0}$. Here, we take into account only the infeasibilities. In this case, the inequality in Step 5 can be ignored.

For $\mathcal{V}_i$, we list two possible options:

- (V1) Simply take $\mathcal{V}_i = M_i s_i$ for all i . This strategy aims to eliminate the most violated constraints that produces a “*local*” reduction on the infeasibilities, although apparently there is no theory ensuring a *global* reduction on the weighted sum of slacks;
- (V2) For each $i \in S \setminus \underline{S}$, we try to solve (P) with $s_i = 0$. If the resulting model is feasible, let us say with solution $(\underline{\mathbf{p}}, \underline{\mathbf{s}})$, we take $\mathcal{V}_i = 1/F(\underline{\mathbf{p}})$. Otherwise, we define $\mathcal{V}_i = -\infty$. This greedy strategy has a more comprehensive view of the effect on the reduction of the infeasibilities. We can expect that each $\mathcal{V}_i$ is computed with few iterations of dual simplex. Thus, we believe that computational cost is reasonable even for real instances.

Finally, another possibility for identifying sets of violated constraints is to use the “conflict refiner” strategy implemented in the IBM© Cplex© package. Cplex is the package currently used in DESSEM software to solve MIPs/LPs. This strategy aims to identify a small set of mutually contradictory constraints. See https://www.ibm.com/support/knowledgecenter/SSSA5P_latest/ilog.odms.cplex.help/CPLEX/UsrMan/topics/infeas_unbd/conflict_refiner/01_ref_confl_title_synopsis.html for details.

References

- [1]CCEE, ONS. “Segundo Relatório de Análise do Programa Piloto de Resposta da Demanda”. NT CCEE 0045/2019, ONS 0061/2019. 2019.
- [2]J. W. Chinneck. *Feasibility and Infeasibility in Optimization: Algorithms and Computational Methods*. Vol. 118. Springer Science & Business Media, 2007.
- [3]J. W. Chinneck and E. W. Dravnieks. “Locating minimal infeasible constraint sets in linear programs”. In: *ORSA Journal on Computing* 3.2 (1991), pp. 157–168.
- [4]O. Guieu and J. W. Chinneck. “Analyzing infeasible mixed-integer and integer linear programs”. In: *INFORMS Journal on Computing* 11.1 (1999), pp. 63–77.
- [5]T. Santos, A. Diniz, C. Saboia, R. Cabral, and L. Cerqueira. “Hourly pricing and day-ahead dispatch setting in Brazil: The "DESSEM" model”. In: *Electric Power Systems Research* 189 (Dec. 2020). Part of special issue: Proceedings of the 21st Power Systems Computation Conference (PSCC 2020), edited by D. Molzahn, p. 106709.
- [6]M. Tamiz, S. J. Mardle, and D. F. Jones. “Detecting IIS in infeasible linear programmes using techniques from goal programming”. In: *Computers & operations research* 23.2 (1996), pp. 113–119.
- [7]J. Vada, O. Slupphaug, T. A. Johansen, and B. A. Foss. “Linear MPC with optimal prioritized infeasibility handling: application, computational issues and stability”. In: *Automatica* 37.11 (2001), pp. 1835–1843.
- [8]J. N. M. van Loon. “Irreducibly inconsistent systems of linear inequalities”. In: *European Journal of Operational Research* 8.3 (1981), pp. 283–288.

Appendix A Other possible approaches

For completeness, we list some alternative methodologies that were discussed during the workshop, but not implemented.

A.1 Minimizing infeasibility using goal programming

Consider the problem (4.1), and suppose that this problem is infeasible. The problem of infeasibility resolution can be formally modeled via multi-objective programming:

$$\begin{aligned} \min \quad & \{s_i, i = 1, 2, \dots, m\} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{p} - \mathbf{s} \leq \mathbf{b}, \end{aligned} \tag{A 1}$$

where m is the number of slacks variable. One widely used method to find the efficient solutions of multi-objective optimization problem is the weighted sum method that combines all the functions into one scalar. The weighted goal program variant allows for direct trade-offs between all unwanted slack variables by placing them in a weighted, normalized single achievement function:

$$\begin{aligned} \min_{\mathbf{s} \geq \mathbf{0}} \quad & \sum_i \frac{w_i s_i}{k_i} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{p} - \mathbf{s} \leq \mathbf{b} \end{aligned} \tag{A 2}$$

where $\sum_i w_i = 1$, $w_i \geq 0$ and k_i is the normalization constant associated with the i -th goal. These constants are necessary in order to scale all the goals onto the same units of

measurement. One possible choice to consider to k_i is known as percentage normalization, where each deviation is turned into a percentage value away from its target level, in this case it is considered $k_i = b_i$.

Based on the empirical experience, experts may be able to establish that with high probability, feasibility is achieved with no more than 5% of deterioration of the displacement measure $F(\mathbf{p})$. Thus, it is reasonable to expect that the goal programming technique obtains an acceptable feasible $\mathbf{p}$ inserting the constraint $F(\mathbf{p}) \leq (1 + \tau)F(\mathbf{p}^*)$ on the goal problem. That is, we focus exclusively on the feasibility maintaining the maximum deterioration as a hard constraint. In this case, F must be easy (perhaps linear) to be handled by solvers. So, option (F1) is indicated for this strategy.

A.2 Using Irreducibly Inconsistent Sets (IIS) and “conflict refiner” algorithms

The identification of a minimal infeasible set of constraints is closely related to the problem of eliminating infeasibilities. Actually, one can argue that the heuristics described in Algorithm 1 is clearly inspired by the techniques described in this section.

There is an extensive literature in finding a *Irreducibly Inconsistent System* (IIS) [3, 4, 6]. It is a minimal constraint set that make the problem infeasible. This is set typically not unique [8].

In [3], the authors propose some algorithms to find at least one IIS in a linear optimization problem. The first idea is a brute force method that consists on eliminating one constraint at a time and resolving the model. If the relaxed model is feasible, then the constraint make it infeasible and must belong to an IIS. This idea is called *Deletion Filter*. However, there is no guarantee that there is a single constraint that once removed will make the model feasible and subsets of the constraints might need to be considered making the method too expensive. When the problems have too many constraints, the authors propose to look for an *Irreducibly Inconsistent Set of Functional Constraints* that consist only on (functional) equalities constraints and ignore bounds. This can be achieved using classical linear algebra techniques [8].

Another approach, from the same authors, is called *Elastic Filter*. It adds slack variables to all constraints, like done for problem (4.2), and solving the relaxed problem trying to minimize the slacks. The constraints associated to positive slacks are saved in the output set and removed from the model that is solved again in an iterative process until it becomes feasible. This idea has the advantage of solving fewer problems as they try to add a set of constraints to the IIS at each step.

The *Additive Method* [6] adds one constraint at a time and solves the resulting system looking for infeasibility. The first constraint that induces an infeasible problem must belong to an IIS, probably together with other constraints that were inserted before. The method adds the constraints that it knows that belong to an IIS at the first steps and trying to identify new IIS constraints. It keeps up with this process until it detects infeasibility only with the constraints that were proved to belong to an IIS.

Such methods were extended to MIP in [4]. Moreover, there are different approaches based on *Model Predictive Control*. The paper [7] uses ideas from *Optimal Weight Design*.

This last work can use weights to pinpoint constraints that are more relevant than others. A good general reference for infeasibility detection is [\[2\]](#).