Recent Answer Set Programming Applications to Scheduling Problems in Digital Health

Paola Cappanera[5], Simone Caruso[1], Carmine Dodaro[2], Giuseppe Galatà[3], Marco Gavanelli[4], Marco Maratea[2,*], Cinzia Marte[2], Marco Mochi[1], Maddalena Nonato[4] and Marco Roma[5]

[1]*DIBRIS, University of Genoa, Genoa, Italy*
[2]*DeMaCS, University of Calabria, Rende, Italy*
[3]*SurgiQ srl, Genova, Italy*
[4]*DE, University of Ferrara, Italy*
[5]*DINFO, University of Firenze, Italy*

Abstract

The Answer Set Programming (ASP) methodology has been recognized to be a viable solution to many practical applications, including scheduling problems in the Healthcare sector, where ASP proved to be an effective solution since 2017. In this paper, we present new scheduling problems in such field that have been successfully solved via ASP in the last few years. The interesting point is that some of these applications either deal with parts of the Healthcare process that were not previously addressed (i.e., the Pre-operative operations), or needed new solving approaches in order to be solved efficiently (i.e., the Chronic Outpatients problem solved via a Logic-Based Bender Decomposition method). For some of the problems discussed, we also provide preliminary experiments not appearing in previous publications on such problems. For all presented problems, we are also working on more "practical" issues, like solving rescheduling-related problems, providing explainability features, and implementing web applications for easy access to such solutions.

Keywords

Answer Set Programming, Scheduling Applications, Digital Health

1. Introduction

Answer Set Programming (ASP) [2, 3] has been recognized to be a viable methodology for solving many practical applications in, e.g. Artificial Intelligence, Bioinformatics, Hydroinformatics, Knowledge Management, and Databases [4, 5, 6, 7, 8, 9]; more recently, ASP has been applied to solve industrial applications [10, 11, 12]. Indeed, the simple syntax [13] and the intuitive semantics, combined with the availability of robust implementations (see, e.g. [14, 15]) make ASP an ideal candidate for addressing combinatorial problems that naturally arise in these contexts, and to combine ASP with machine learning approaches (see, e.g., [16, 17]).

More recently, since 2017, ASP has been also applied to solving scheduling problems, and in particular problems in the healthcare domain. The advent of the COVID-19 pandemic in 2020 significantly improved the interest to the deployment on automated solutions based on Artificial Intelligence techniques in Healthcare.

AI4CC-IPS-RCRA-SPIRIT 2024: International Workshop on Artificial Intelligence for Climate Change, Italian Workshop on Planning and Scheduling, RCRA Workshop on Experimental evaluation of algorithms for solving problems with combinatorial explosion, and SPIRIT Workshop on Strategies, Prediction, Interaction, and Reasoning in Italy. November 25-28th, 2024, Bolzano, Italy [1].

[*]Corresponding author.
[†]These authors contributed equally.
paola.cappanera@unifi.it (P. Cappanera); simone.caruso@edu.unige.it (S. Caruso); carmine.dodaro@unical.it (C. Dodaro); giuseppe.galata@surgiq.com (G. Galatà); marco.gavanelli@unife.it (M. Gavanelli); marco.maratea@unical.it (M. Maratea); cinzia.marte@unical.it (C. Marte); marco.mochi@edu.unige.it (M. Mochi); maddalena.nonato@unife.it (M. Nonato); marco.roma@unifi.it (M. Roma)
0000-0003-3674-3896 (P. Cappanera); 0000-0002-2724-4342 (S. Caruso); 0000-0002-5617-5286 (C. Dodaro); 0000-0002-1948-4469 (G. Galatà); 0000-0001-7433-5899 (M. Gavanelli); 0000-0002-9034-25276 (M. Maratea); 0000-0003-3920-8186 (C. Marte); 0000-0002-5849-3667 (M. Mochi); 0000 0001 8708 120X (M. Nonato); 0000-0002-0925-212X (M. Roma)

In this paper, we present recently solved scheduling problems in such field that have been successfully solved via ASP in the last few years. The problems we will present with are: the *Pre-operative Assessment Clinic* (see, e.g. [18, 19, 20]), which deals with the phases before the admission to the Hospital for the surgical procedure; the *Periodic Treatments* (see, e.g. the surveys [21, 22]), consisting of planning a care path over a period of several weeks, in which patients have to perform different treatments respecting a certain periodicity; the *Nuclear Medicine* [23, 24, 25], which accounts for scheduling patients that must use radiopharmaceuticals; the *Non-communicable Chronic Diseases Agenda*, which consists of scheduling periodic health services for a set of chronic patients with co-morbidity [26]; and the *mid-term scheduling of surgeons in a hospital network*. The interesting point is that some of these applications either deal with parts of the Healthcare process that were not previously addressed with ASP (i.e., the Pre-operative operations), or needed new solving approaches in order to be solved efficiently (i.e., the Non-communicable Chronic Diseases Agenda problem solved via a Logic-Based Benders Decomposition method [27]). For some of the problems discussed, i.e., Nuclear Scheduling and the Non-communicable Chronic Diseases Agenda, we also provide preliminary experiments or new analyses not appearing in previous publications on such problems. For all presented problems, we are also working on more "practical" issues, like solving rescheduling-related problems (see, e.g. [28]), providing explainability features (see, e.g. [29] for a preliminary application in the context of Operating Room scheduling), and implementing web applications for easy access to such solutions (see, e.g. [28]).

The paper is structured as follows. Section 2 describes the afore-mentioned problems, while Section 3 shows novel experiments for some of such problems. Finally, Section 4 outlines further related problems that have been solved with ASP in this context, and Section 5 concludes the paper and envisages directions for future research.

2. Application Descriptions

In this section, we describe the most recent ASP applications in the Healthcare domain. One subsection is devoted to each application.

2.1. Pre-operative Assessment Clinic

The Pre-operative Assessment Clinic (PAC) scheduling problem is the task of assigning registrations to a day on which patients will undergo a series of pre-surgical exams. The complexity of the problem arises from the need to account for multiple aspects, including different priority levels for each registration, due dates, and operators' availability. The PAC problem is divided into two-subproblems by many hospitals, including the one from which we derived the specification of the problem. The first one focuses on assigning the registrations to a day, linking the registrations with a default list of exams based on the surgical specialty of their procedure. The schedule must respect due dates, which indicate the latest possible day a patient can be scheduled, and target days, which represent the optimal days for scheduling, while prioritizing patients with higher urgency. The exams are conducted in specific exam areas, each requiring an operator to be activated and having a limited available usage time. Operators can activate up to three different exam areas but can be assigned to only one area per day. The solution to the first sub-problem involves assigning both the operators to the exam areas and the PAC day for the patients, ensuring that the total usage time of each exam area remains within its limit. In this problem, it is important to ensure that it will be possible to treat all the patients that are assigned to a date, even if the final list of required exams is not known. Thus, in the first sub-problem the solution assigns patients overestimating the duration and the number of exams needed. In particular, all the optional exams, such as exams required by smokers or patients with diabetes, are assigned to all the patients in the first phase. In a later stage, when the operation day is closer, the hospital defines the final list of exams and it is possible to assign the correct starting time for all the needed exams. Thus, in the second sub-problem, the solution has to assign the starting time of each exam, having the first sub-problem already assigned the day. The input consists of registrations, exams needed by patients, and the exam areas activated. Exams are ordered, so the solution must assign the starting time of each exam respecting their order and their duration, by considering that each exam area can be used by one patient at a time. Finally, the solution minimizes the difference between the starting time of the first exam and the last exam of each patient. Solutions to this problem employing ASP can be found in [28, 30].

2.2. Periodic Treatments

The Periodic Treatments scheduling problem (see [31] for a solution based on ASP) consists of assigning recurring appointments over multiple weeks according to a specific care path. A typical use case is that, after a medical examination, the patient comes with a precise list of treatments, e.g. 10 sessions of physiotherapy followed by 10 minutes of cardio training, and asks for the scheduling of a certain number of appointments based on their availability and on the availability of the care facility. Thus, a care path includes several sessions that must be scheduled according to certain constraints: the weekly-based periodicity, such as weekly or biweekly, and a minimum and/or maximum number of sessions per week.

Moreover, a session can require multiple treatments and, in this case, they are typically scheduled on the same day, one immediately after the other, so that the patient can come just once for all of them. Treatments may also require different instruments or types of machinery and operators, thus the resources must be properly allocated to patients.

Operators can also be assigned to different tasks during the day, therefore they can be assigned to a machinery or instrument only if their tasks allow, and they may not be necessary for the total duration of the treatments, e.g. the patient starts the machinery and then does other tasks. It is also possible that the care facility operates in multiple locations: in this case, the operators can move during the day if necessary and we have to take into account the travel time while scheduling the appointments.

Finally, patients may also have preferences for specific days or times for their appointments, as well as times and days when they are not available.

An optimal solution assigns the patients' appointments on their desired days of the week and desired starting times, minimizes the waiting times between treatments within a session and the care path begins as close as possible to the target week prescribed by the doctor. The operators' working time is daily maximized and weekly balanced among the available operators. This can be achieved through different strategies, such as minimizing the daily working operators or reducing travel time when operators are required to work at multiple locations.

As said above, the main use case consist in a patient who calls a healthcare provider and receives a list of appointments. Then the patient can accept the proposed appointments or reject them and negotiate for other times, dates, and locations. However, another use case is when the healthcare provider collects a waiting list of patients and assigns their care paths at the same time. This approach has the advantage of being able to better optimize the working times of the operators and the instruments, although it makes the negotiating phase with the patients more difficult.

2.3. Nuclear Medicine

The Nuclear Medicine Scheduling (NMS) problem focuses on the challenge of efficiently scheduling patients for specific days, assigning them to a tomograph and/or an injection chair, depending on the specific needs of the patients or the requirements of the procedure. Each procedure is associated with a specific protocol, which is composed of several distinct phases. These phases typically include anamnesis (the process of collecting the medical history of the patient), a medical check (an evaluation of the patient's current health status), injection of radiopharmaceuticals followed by a biodistribution period (the time required for the radiopharmaceuticals to spread throughout the body), and image detection (capturing medical images using a tomography). Furthermore, the duration of each of these phases is not fixed but depends on both the type of exam being conducted and the protocol being followed [32].

This variability introduces a layer of complexity to the scheduling process, as multiple phases must be coordinated across a variety of protocols. Moreover, each protocol may require different resources, times, and constraints, further complicating the task of developing an optimal schedule. A suboptimal scheduling not only increases the costs of medical equipment but is particularly harmful to patients. For instance, the timing and sequence of certain phases, such as the injection and the biodistribution of radiopharmaceuticals, are critical for accurate image detection. Furthermore, clinics may vary in terms of available resources, such as rooms, tomographs, and injection chairs. These constraints, combined with differing protocols, mean that each clinic may face different challenges in computing an optimal schedule.

The goal is to assign patients to time slots within a single day, maximizing resource utilization while minimizing delays and patient wait times. In general, a proper solution must satisfy the following conditions: (i) each scheduled patient must have a defined start and end time for every required phase, (ii) no more than two patients can undergo the medical check phase simultaneously, (iii) the injection phase must take place in an injection chair or on a tomograph, depending on the protocol, and (iv) each injection chair and tomograph can only be used by one patient at a time. Moreover, the solution aims at maximizing the number of patients that can be scheduled within the clinic's available resources. At the same time, the solution should minimize the amount of unnecessary time patients spend in the clinic, such as waiting for their next phase, since the reduction of patient wait times not only improves patient satisfaction but also enhances the overall effectiveness of the medical exams, ensuring that each phase is completed in a coordinated manner.

2.4. Non-communicable Chronic Diseases Agenda

As life expectancy increases in the western world, the number of elderly people is also steadily increasing. A large percentage of elderly people suffer from several diseases that are noncommunicable, such as hypertension, diabetes, obesity, just to name a few. Well-established clinical pathways are available for most of the Non-communicable Chronic Diseases (NCDs); they typically require the patient to receive treatments or medications periodically, and, as diseases are often chronic, patients must adhere to the cures for the rest of their life. Often such treatments must be given in a hospital, but, since they are required only for a few hours, hospitalization is avoided as much as possible, with the twofold aim of letting the patients live at home with their dear ones and to avoid occupying an expensive hospital bed.

In addition, elderly patients often suffer from several diseases at the same time (called comorbidity), so they have to follow several pathways, including different treatments, to be provided with different periodicities, with cures that can be conflicting. For example, if one pathway requires the patient to take a drug, the effect could invalidate a blood sample analysis (prescribed by another pathway) for a specific time interval. This is an example of an incompatibility between treatments: one treatment might invalidate or interact (in a positive or negative way) with another treatment. In other cases, one service requires another treatment to be performed before or within some deadline; for example, a visit with a specialist doctor might require that some analysis have been performed before, and the results must be available at the visit.

The interactions between services (within a pathway or involving two different pathways) generate a complex combinatorial problem, that a fragile patient might be unable to solve if left alone. For this reason, we proposed a centralized approach [33], managing all the pathways of the outpatients receiving services in a hospital. Given a set of patients (each with a clinical pathway), a set of time availabilities of services at the hospital, the NCD problem consists in assigning to each service required by a pathway a day and a time, together with an available operator such that (i) a patient does not have two different services scheduled at the same time, (ii) an operator cannot serve two patients at the same time, (iii) each service is provided with the periodicity required by the clinical pathway (possibly, with some given tolerance), and (iv) in case the problem is unsolvable, some services can be delivered in a private clinic, but at a higher cost for the national health service, so the number of services delivered outside the hospital should be minimized.

In order to improve the scalability of the approach, we proposed a decomposition of the whole problem [33] into a master problem (which assigns to each service scheduled in the hospital a day) and a sub-problem for each day (which assigns an operator and a time to each service scheduled within the day). The encoding of the master problem is provided in [34]. In [33], the master problem proposes to the sub-problems a set of services to be scheduled in the hospital, but the sub-problem might find impossible to assign a time to all the services, so some service might be assigned to a private clinic. The decomposition improves significantly the scalability of the approach, but does not guarantee the optimality of the obtained solution: another solution of the master problem could result in a better assignment, with a higher number of services provided by the hospital. For this reason, in [26] we developed the first Logic-Based Benders Decomposition (LBBD) approach in ASP; LBBD guarantees optimality (given enough time), while usually providing a good performance, due to the decomposition approach.

2.5. Mid-term scheduling of medical staff at a hospital network

ASP was also applied to the scheduling of medical staff (in particular, surgeons) in a network of clinics; surgeons have to provide services in a main hospital, plus a network of smaller hospitals in nearby villages. The distribution of services in several locations clearly poses the obvious problem of allowing enough time to let surgeons travel from one location to another.

Moreover, the hospital faced several reorganizations due to several factors, including the recent pandemic, the varying availability of medical staff, and the varying number of provided services. These changes have exacerbated previous strains to provide working schedules able to cover the service, meet legal requirements, and be perceived as a fair workload allocation.

The scheduling was previously computed by hand by some of the doctors, who had to do it for free, outside working hours. Beside being a long and unpleasant task, completely outside the skills for which a medical doctor is trained, it was also highly error-prone, and several times in the past errors were found at the last moment, with need to recover the error during emergency situations.

The model of the problem shares similarity with classical scheduling problems; however, while most of the tasks are assigned on a weekly basis (short-term), others are assigned on a monthly basis (mid-term). For example, a number of doctors must be available during the week-end, and the same doctor cannot be assigned more than two week-ends in a month.

An important issue is fairness: among the several tasks in the hospital, some are preferred by the doctors, while others are less preferred; for example, beside surgery in the operating room, outpatient activity is carried out in the various locations of the hospital (for example, proctology clinic), so it is important to rotate the tasks amongst all the available personnel. On the other hand, not all doctors are allowed to perform all tasks, due to different specialization in the medicine field, but also to illnesses, conditions, or other regulations (e.g., pregnant women and breastfeeding mothers cannot be assigned to night shifts).

A naïve approach would be to divide the less preferred tasks in an equal share amongst all the doctors, but this would conflict with the fact that some doctors can perform many different tasks, while others are limited to a restricted set of tasks. Moreover, it is impossible to balance the assignment of some of the tasks in a week, as there are tasks executed only once a week: the balancement must be necessarily obtained in the long term. In addition, doctors should be assigned to the largest variety of activities allowed by their skills, to keep them trained on all of them.

To complicate the problem, doctors are routinely asked to work overtime. The overtime is paid slightly more than a regular working hour, but not enough to make it attractive. Again, fairness in assigning overtime becomes an important issue.

We developed an ASP-based solution for this problem, with a rolling-horizon approach: the ASP solver is run for a sequence of weeks. After obtaining the solution for one week, the assignment is saved into a long-term memory, so that the ASP program in the following weeks can take into consideration the assignment in the previous weeks. This is important to satisfy hard constraints on the mid-term (e.g., no three weekends in a month) and also to balance the solution in the long term (even for tasks executed once a week, at the end of the year they should be divided in a fair way to the available doctors).

Finally, the hospital often faces reconfigurations (due to the pandemics, but also due to the fact

Table 1
Results scheduling one patient at a time.

Number of treatments	BASE	BASE_OPT	MS	MS_OPT
12 treatments (in 6 appointments)	5.21s	3.97s	1.54s	1.07s
18 treatments (in 7 appointments)	7.7s	6.22s	2.44s	1.82s
24 treatments (in 12 appointments)	11.83s	9.78s	5.25s	5.83s
27 treatments (in 9 appointments)	11.63s	9.53s	4.07s	3.59s
36 treatments (in 12 appointments)	17.74s	14.31s	8.4s	7.66s

that some doctor might become unavailable for maternity, illness, etc.); in reconfigurations, some services have been closed or some services are opened. The presence of an automated scheduling tool lets the medical staff perform what-if analysis, checking how the schedule would change in case of reconfigurations, and whether introducing new services would be sustainable with the current staffing level.

In future work, the assignment could be performed in a hierarchical way, with a monthly assignment that schedules the tasks with a mid-term periodicity, followed by several weekly assignments that schedule the remaining tasks. Such configuration could reduce the computation time and possibly provide schedules with higher fairness for each week.

Another future work concerns addressing the rescheduling problem: currently, in case of major disruptions (such as unavailability of a doctor), the schedule is rebuilt from scratch, possibly changing completely the schedule at a short notice. A better approach could change the objective function, and try to find a feasible schedule (fulfilling the new constraints) with as few modifications as possible from the previous schedule.

3. Experimental Results

In this section we present some (preliminary) experimental results for two of the problems previously described.

3.1. Experiments on Periodic Treatments

In [31], we introduced an ASP encoding for the Periodic Treatments scheduling problem and conducted a preliminary experimental analysis that showed that ASP is a suitable solution for this problem. We then developed an enhanced ASP encoding which improved the performance. More in detail, we made multi-shot-based encoding that tries to assign the start of the care path in the optimal week (i.e., the one requested by the doctor), then, if a solution is found, that is our optimal solution and we stop, otherwise, we increment and decrement the target week by one in each iteration until a solution is found. This approach reduces (in most cases) the time spent in the grounding phase, as fixing the starting week narrows down the possible combinations. Moreover, with enough computational resources, the problem can be solved in parallel as each sub-problem is independent. Additionally, the search space can be further reduced by, in a pre-processing phase, computing the time slots in which treatments cannot be assigned, such as when all the operators are unavailable or instruments are already occupied. The experiments were run on a AMD Ryzen 5 3600 CPU @ 3.60GHz with 16 GB of physical RAM. As ASP system, we used CLINGO version 5.4.1, configured with the option –parallel-mode 6 for parallel execution. The time limit was set to 60 seconds for the scheduling of single patients, and 300 seconds for the scheduling of multiple patients. Table 1 shows a comparison of the base encoding as presented in [31] (BASE), the base encoding with the pre-processing optimization (BASE_OPT), the multi-shot encoding (MS), the multi-shot encoding with the pre-processing optimization (MS_OPT). The data used in this experiment were randomly generated. In particular, we generated and scheduled one patient at a time, up to one hundred patients, considering a period of 10 weeks, each with 7 working days and, for each day, 60 time slots. The facility was in two locations, with 4 operators that could move between

Table 2
Results scheduling lists of patients.

N.Pat/instance	Avg.Treat/instance	Avg.Time	%OPT
5 patients	57	34.61s	100%
10 patients	122	191.3s	70%
5 patients	92	61.12s	100%
10 patients	186	237s	40%

them and one facility had 3 instruments while the other 2. Each patient required between 2 to 3 weekly sessions, and the optimal week, in which the care path starts, was set in the first 4 weeks of the planning period. The number of treatments per session, instead, varied between 2 and 3 treatments and with a duration ranging from 3 to 12 time slots. The optimization on the research space produces a slight improvement in the average times due to the reduced search space. However, what makes the encoding definitely perform better is the multi-shot strategy by which the grounding is reduced significantly. The idea is to avoid the computation of all the possible assignable weeks, but to add an assignable week at each iteration until a solution is found. This strategy combined with the optimization of the time slots allowed us to get times more than halved and to solve many instances in a few seconds, which is ideal for real-world scheduling where it is crucial to have solutions in a short time.

Finally, another scenario is when the healthcare provider schedules multiple patients simultaneously. The results are shown in Table 2. We generated instances with 5 and 10 patients, until scheduling one hundred patients, varying the average number of total treatments per instance, i.e., the average number of treatments per patient. For this scenario, we do not have an actual comparison with the base encoding and the result presented in [31], as this approach improved performance at the cost of optimality (column "%OPT" which indicates the percentage of instances for which the optimum is found) which was, instead, guaranteed in the previous version. However, the results show better scalability with lower average times and higher percentages of optimum still guaranteeing a sub-optimal solution for all the instances tested. We have gone as far as instances with 20 patients, although, we could only get sub-optimal solutions before the time out. This strategy consists of assigning as many patients as possible in the target week and gradually incrementing and decrementing it if necessary, similar to the single-patient strategy explained above. Despite this approach does not guarantee the optimum, performance improvement can be a good trade-off for practical usage.

3.2. Experiments on Non-communicable Chronic Diseases Agenda

In [26], we addressed the NCD agenda problem and compared three alternative approaches; note again that these three approaches (differently from [33]) are complete methods, i.e., they are able to obtain the true optimal solution if enough time is allotted. The three approaches are:

- a monolithic approach, in which there is only one ASP program addressing the whole problem,
- an approach based on LBBD, in which both the master problem and the sub-problem are solved through ASP programs; the two programs are iteratively invoked by an external script,
- a LBBD approach exploiting the multi-shot solving feature of clingo [35] to avoid restarting the solution process from scratch in each iteration.

The instances were randomly generated with a realistic simulator, based on publicly available medical guidelines for the most common NCDs.

In these experiments, we consider: 5 Care Units, each having a daily capacity drawn with uniform probability in the range [24, 60] (in generic time units); for each Care Unit, a number of operators drawn with uniform probability in [1,4] on each day of the week is available; the duration of services is randomly drawn in [6,15] and associated with a randomly selected Care Unit; packets made of 4 services at most; a given number of patients, each with a number of pathways in [1,4], with probability

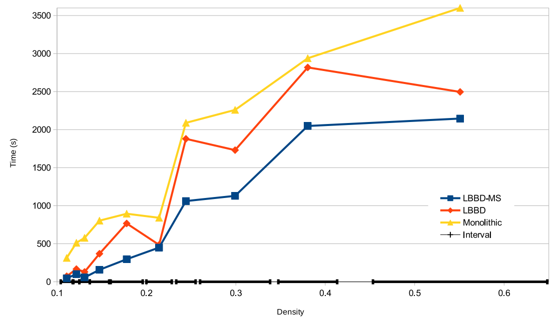

Figure 1: NCD Agenda problem: running time of a Monolithic approach, LBBD and LBBD with Multi-Shot solving (LBBD-MS) varying the density of requested services.

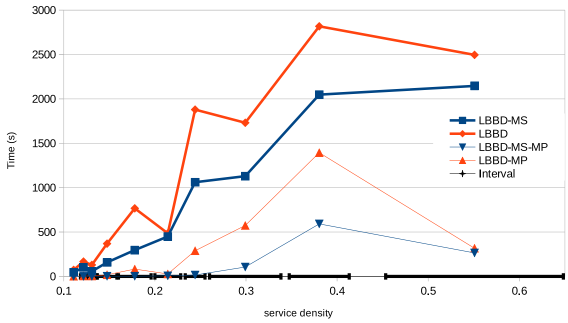

Figure 2: NCD Agenda problem: running time of LBBD (with and without MultiShot, MS) varying the service density. The running time of the Master Problem (MP) is drawn in a thin line, while the total running time is in a thick line.

inversely proportional to the number of pathways. For each number of patients in $\{10, 20, 40\}$ and length of the planning horizon in {30,60} days, 20 instances are generated, summing up to 120 instances.

Experiments were run with Clingo 5.6.2 with a time limit of one hour on a Ubuntu 22.04.1 LTS OS, Intel(R) Xeon(R) CPU E5-2430 v2 @ 2.50GHz machine with 32GiB RAM.

In order to test the scalability of the approach, we sorted the instances by density of the requested services, where the density is defined as the average number of services requested by a patient in each day, i.e.:

$$density = \frac{|Services|}{|Days||Patients|}$$

where $Services$ is the set of services, $Days$ the set of days in the horizon, and $Patients$ is the set of patients. In Figure 1, we show the running time varying the density of services; we grouped the instances in groups of ten, and took the geometric mean of the running time. In case an algorithm could not solve to optimality the program (e.g., due to out of memory), we assign it the timeout (3600s). Clearly, as the density of services increases, the instances become harder, up to the point that for the monolithic approach all instances with a density over 0.45 could not be solved within the timeout (see Fig. 3). Clearly, the LBBD approaches are significantly faster than the monolithic approach, with Multi-Shot that provides an additional speedup.

In Figure 2 we compare the two implementations of LBBD and also plot the time required to solve the master problem. From the graph, the running time of the Master Problem grows up to a density around 0.4, then it decreases, showing that the Sub-Problems take a significant time.

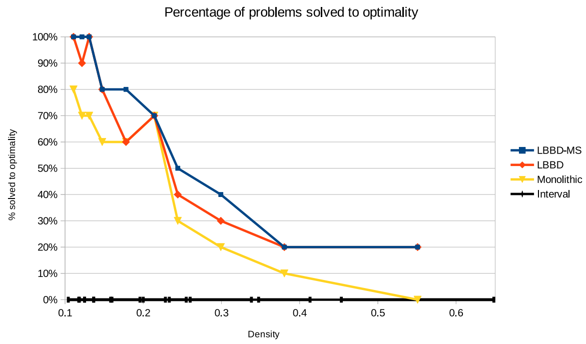

Figure 3: NCD Agenda problem: percentage of instances that could be solved to optimality within 3600s.

4. Related Work

As we previously mentioned, ASP has been successfully used for solving hard combinatorial and application scheduling problems in several research areas. In the healthcare domain, the first solved problem was the *Nurse Scheduling Problem* [36, 37], where the goal is to create a scheduling for nurses working in hospital units. Then, the problem of assigning Operating Rooms to patients, denoted as *Operating Room Scheduling*, has been treated [38], further extended to include bed management [39] and adapted to real data [40]. Further problems include the *Chemotherepy Treatment Scheduling* problem [41], in which patients are assigned a chair or a bed for their treatments, and the *Rehabilitation Scheduling Problem* [42], which assigns patients to operators in rehabilitation sessions. For all the works mentioned, also rescheduling solutions have been defined, implemented and tested [43, 44, 41], and for many of them web applications, in which an user can run a solution without the need of local installations, have been provided.

The authors of [45] schedule multi appointments for rheumatic outpatients at a Hospital Day Service, where multidisciplinary diagnostic tests and therapies are delivered. Service capacity is known and deterministic. The problem is encoded into ASP. Patients are partitioned into three classes with decreasing priority; to reduce computing time the schedule is computed separately for each priority class.

Out of the healthcare domain, El-Kholany et al. [46] present a decomposition scheme in ASP for Job Shop Scheduling, driven by a machine learning algorithm. There is no feedback from the Sub-problem to the Master-problem, each Sub-problem is solved only once, so the resulting algorithm cannot prove optimality of the found solution (it is a heuristic algorithm). The approach is further improved in [47]. Francescutto et al. [48] solve a variant of the job-shop scheduling problem using Multi-Shot Solving to perform a sequence of runs of an ASP solver enhanced with Difference Logic; in the runs they change the bounds to avoid solving a problem with a large grounding in a single run.

5. Conclusion

In this paper, we have over-viewed recent ASP applications for scheduling problems in Digital Health. Some of the applications deal with parts of the healthcare process not previously dealt with ASP, or needed alternative solving methods in order to produce efficient solutions. For two of the outlined applications, we have also presented very recent, unpublished experimental results. As future work, we first plan to expand this survey and present a complete picture of all ASP applications for scheduling problems in Digital Health. Then, given that many of our ASP solutions have confronted to other logic-based formalisms, such as SAT, Max-SAT, ILP, and PB on automated translation of ASP instances, we would like to extend this analysis to all solutions and include in the picture also SMT solvers (see, e.g., [49, 50]) when applicable.

6. Acknowledgments

Research funded by the Italian Ministry of University and Research through PNRR - M4C2 - Investimento 1.3 (Decreto Direttoriale MUR n. 341 del 15/03/2022), Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 8 "Pervasive AI", funded by the European Union under the NextGeneration EU programme".

This research has been also partially supported by Italian Ministry of Research (MUR) (i) under PNRR project FAIR “Future AI Research”, CUP H23C22000860006, (ii) with the European Union - NextGenerationEU, National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.5, project “RAISE - Robotics and AI for Socio-economic Empowerment” (ECS00000035) under the project “Gestione e Ottimizzazione di Risorse Ospedaliere attraverso Analisi Dati, Logic Programming e Digital Twin (GOLD)”, CUP H53C24000400006, and (iii) under PRIN project PINPOINT, CUP H23C22000280006; and by Italian Ministry of Enterprises and Made in Italy (MIMIT) under project EI-TWIN “Concezione, progettazione e sviluppo di soluzioni di Energy digital TWIN” Prog n. F/310168/05/X56 - CUP: B29J24000680005 - COR: 22338845. The research of Giuseppe Galatà is partially funded by the ”POR FESR Liguria 2014-2020” public grant scheme. Carmine Dodaro and Marco Gavanelli are members of Gruppo Nazionale Calcolo Scientifico (GNCS) of INdAM.

References

[1] D. Aineto, R. De Benedictis, M. Maratea, M. Mittelmann, G. Monaco, E. Scala, L. Serafini, I. Serina, F. Spegni, E. Tosello, A. Umbrico, M. Vallati (Eds.), Proceedings of the International Workshop on Artificial Intelligence for Climate Change, the Italian workshop on Planning and Scheduling, the RCRA Workshop on Experimental evaluation of algorithms for solving problems with combinatorial explosion, and the Workshop on Strategies, Prediction, Interaction, and Reasoning in Italy (AI4CC-IPS-RCRA-SPIRIT 2024), co-located with 23rd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2024), CEUR Workshop Proceedings, CEUR-WS.org, 2024.

[2] M. Gelfond, V. Lifschitz, The stable model semantics for logic programming, in: Proceedings of the Fifth International Conference and Symposium , Seattle, Washington, August 15-19, 1988 (2 Volumes), MIT Press, 1988, pp. 1070–1080.

[3] M. Gelfond, V. Lifschitz, Classical Negation in Logic Programs and Disjunctive Databases, New Generation Comput. 9 (1991) 365–386.

[4] M. Balduccini, M. Gelfond, R. Watson, M. Nogueira, The USA-Advisor: A case study in answer set planning, in: LPNMR, volume 2173 of *LNCS*, Springer, 2001, pp. 439–442.

[5] E. Erdem, U. Öztok, Generating explanations for biomedical queries, Theory and Practice of Logic Programming 15 (2015) 35–78.

[6] L. Koponen, E. Oikarinen, T. Janhunen, L. Säilä, Optimizing phylogenetic supertrees using answer set programming, TPLP 15 (2015) 604–619.

[7] G. Grasso, S. Iiritano, N. Leone, F. Ricca, Some DLV applications for knowledge management, in: E. Erdem, F. Lin, T. Schaub (Eds.), Proc. of the 10th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2009), volume 5753 of *LNCS*, Springer, 2009, pp. 591–597.

[8] M. C. Marileo, L. E. Bertossi, The consistency extractor system: Answer set programs for consistent query answering in databases, Data Knowl. Eng. 69 (2010) 545–572.

[9] M. Gavanelli, M. Nonato, A. Peano, An ASP approach for the valves positioning optimization in a water distribution system, Journal of Logic and Computation 25 (2015) 1351–1369.

[10] M. Abseher, M. Gebser, N. Musliu, T. Schaub, S. Woltran, Shift design with answer set programming, Fundamenta Informaticae 147 (2016) 1–25.

[11] C. Dodaro, P. Gasteiger, N. Leone, B. Musitsch, F. Ricca, K. Schekotihin, Combining answer set programming and domain heuristics for solving hard industrial problems (application paper), Theory and Practice of Logic Programming 16 (2016) 653–669.

[12] C. Dodaro, N. Leone, B. Nardi, F. Ricca, Allotment problem in travel industry: A solution based on ASP, in: RR, volume 9209 of *LNCS*, Springer, 2015, pp. 77–92.
[13] F. Calimeri, W. Faber, M. Gebser, G. Ianni, R. Kaminski, T. Krennwallner, N. Leone, M. Maratea, F. Ricca, T. Schaub, ASP-Core-2 input language format, Theory and Practice of Logic Programming 20 (2020) 294–309.
[14] M. Alviano, C. Dodaro, Anytime answer set optimization via unsatisfiable core shrinking, Theory and Practice of Logic Programming 16 (2016) 533–551.
[15] M. Gebser, B. Kaufmann, R. Kaminski, M. Ostrowski, T. Schaub, M. T. Schneider, Potassco: The Potsdam answer set solving collection, AI Commun. 24 (2011) 107–124.
[16] P. Bruno, F. Calimeri, C. Marte, M. Manna, Combining deep learning and ASP-based models for the semantic segmentation of medical images, in: S. Moschoyiannis, R. Peñaloza, J. Vanthienen, A. Soylu, D. Roman (Eds.), Proceedings of the 5th International Joint Conference on Rules and Reasoning (RuleML+RR 2021), volume 12851 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 95–110.
[17] P. Bruno, F. Calimeri, C. Marte, Dedudeep: An extensible framework for combining deep learning and asp-based models, in: G. Gottlob, D. Inclezan, M. Maratea (Eds.), Proceedings of the 16th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2022), volume 13416 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 505–510.
[18] M. P. Harnett, D. Correll, S. Hurwitz, A. Bader, D. Hepner, Improving Efficiency and Patient Satisfaction in a Tertiary Teaching Hospital Preoperative Clinic, Anesthesiology 112 (2010) 66–72.
[19] M. Ferschl, A. Tung, B. Sweitzer, D. Huo, D. Glick, Preoperative Clinic Visits Reduce Operating Room Cancellations and Delays, Anesthesiology 103 (2005) 855–859.
[20] D. Correll, A. Bader, M. Hull, C. Hsu, L. Tsen, D. Hepner, Value of Preoperative Clinic Visits in Identifying Issues with Potential Impact on Operating Room Efficiency, Anesthesiology 105 (2006) 1254–1259.
[21] J. Marynissen, E. Demeulemeester, Literature review on multi-appointment scheduling problems in hospitals, European Journal of Operational Research 272 (2019) 407–419.
[22] P. Hulshof, N. Kortbeek, R. Boucherie, E. W. Hans, P. J. M. Bakker, Taxonomic classification of planning decisions in health care: a structured review of the state of the art in OR/MS, Health Systems 1 (2012) 129–175.
[23] E. Pérez, L. Ntaimo, W. E. Wilhelm, C. Bailey, P. McCormack, Patient and resource scheduling of multi-step medical procedures in nuclear medicine, IIE Transactions on Healthcare Systems Engineering 1 (2011) 168–184. doi:10.1080/19488300.2011.617718.
[24] Q. Xiao, L. Luo, S. Zhao, X. bin Ran, Y. bing Feng, Online appointment scheduling for a nuclear medicine department in a chinese hospital, Computational and Mathematical Methods in Medicine 2018 (2018). URL: https://api.semanticscholar.org/CorpusID:5061754.
[25] F. Akhavizadegan, J. Ansarifar, F. Jolai, A novel approach to determine a tactical and operational decision for dynamic appointment scheduling at nuclear medical center, Computers & Operations Research 78 (2017) 267–277.
[26] P. Cappanera, M. Gavanelli, M. Nonato, M. Roma, Logic-based Benders decomposition in answer set programming for chronic outpatients scheduling, Theory and Practice of Logic Programming 23 (2023) 848–864. doi:10.1017/S147106842300025X.
[27] J. N. Hooker, G. Ottosson, Logic-based Benders decomposition, Mathematical Programming 96 (2003) 33–60.
[28] S. Caruso, G. Galatà, M. Maratea, M. Mochi, I. Porro, Scheduling pre-operative assessment clinic with answer set programming, Journal of Logic and Computation 34 (2023) 465–493. URL: https://doi.org/10.1093/logcom/exad017. doi:10.1093/logcom/exad017.
[29] R. Bertolucci, C. Dodaro, G. Galatà, M. Maratea, I. Porro, F. Ricca, Explaining ASP-based operating room schedules, in: R. D. Benedictis, M. Maratea, A. Micheli, E. Scala, I. Serina, M. Vallati, A. Umbrico (Eds.), Proceedings of the 9th Italian workshop on Planning and Scheduling (IPS'21) and the 28th International Workshop on "Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion" (RCRA'21) with CEUR-WS co-located with 20th International

Conference of the Italian Association for Artificial Intelligence (AIxIA 2021), Milan, Italy (virtual), November 29th-30th, 2021, volume 3065 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021. URL: https://ceur-ws.org/Vol-3065/paper7_182.pdf.
[30] S. Caruso, G. Galatà, M. Maratea, M. Mochi, I. Porro, Scheduling pre-operative assessment clinic via answer set programming, in: R. D. Benedictis, M. Maratea, A. Micheli, E. Scala, I. Serina, M. Vallati, A. Umbrico (Eds.), Proceedings of the 9th Italian workshop on Planning and Scheduling (IPS'21) and the 28th International Workshop on "Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion" (RCRA'21), volume 3065 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021.
[31] S. Caruso, C. Dodaro, G. Galatà, M. Maratea, Scheduling periodic treatments via answer set programming, in: A. Dovier, A. Formisano (Eds.), Proceedings of the 38th Italian Conference on Computational Logic, Udine, Italy, June 21-23, 2023, volume 3428 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3428/paper7.pdf.
[32] C. Dodaro, G. Galatà, C. Marte, M. Maratea, M. Mochi, Nuclear medicine scheduling via answer set programming, in: E. D. Angelis, M. Proietti (Eds.), Proceedings of the 39th Italian Conference on Computational Logic (CILC 2024), volume 3733 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024.
[33] P. Cappanera, M. Gavanelli, M. Nonato, M. Roma, Decomposition approaches for scheduling chronic outpatients' clinical pathways in answer set programming, J. Log. Comput. 33 (2023) 1851–1871. URL: https://doi.org/10.1093/logcom/exad038. doi:10.1093/LOGCOM/EXAD038.
[34] P. Cappanera, M. Gavanelli, M. Nonato, M. Roma, A decomposition approach to the clinical pathway deployment for chronic outpatients with comorbidities, in: L. Amorosi, P. Dell'Olmo, I. Lari (Eds.), Optimization in Artificial Intelligence and Data Sciences, Springer International Publishing, Cham, 2022, pp. 213–226.
[35] M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub, Multi-shot ASP solving with clingo, Theory Pract. Log. Program. 19 (2019) 27–82.
[36] C. Dodaro, M. Maratea, Nurse scheduling via answer set programming, in: LPNMR, volume 10377 of *LNCS*, Springer, 2017, pp. 301–307.
[37] M. Alviano, C. Dodaro, M. Maratea, An advanced answer set programming encoding for nurse scheduling, in: AI*IA, volume 10640 of *LNCS*, Springer, 2017, pp. 468–482.
[38] C. Dodaro, G. Galatà, M. Maratea, I. Porro, Operating room scheduling via answer set programming, in: AI*IA, volume 11298 of *LNCS*, Springer, 2018, pp. 445–459.
[39] C. Dodaro, G. Galatà, M. K. Khan, M. Maratea, I. Porro, An ASP-based solution for operating room scheduling with beds management, in: P. Fodor, M. Montali, D. Calvanese, D. Roman (Eds.), Proceedings of the Third International Joint Conference on Rules and Reasoning (RuleML+RR 2019), volume 11784 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 67–81.
[40] C. Dodaro, G. Galatà, M. Gebser, M. Maratea, C. Marte, M. Mochi, M. Scanu, Operating Room Scheduling via Answer Set Programming: improved encoding and test on real data, Journal of Logic and Computation (2024) exae041. doi:10.1093/logcom/exae041.
[41] C. Dodaro, G. Galatà, A. Grioni, M. Maratea, M. Mochi, I. Porro, An ASP-based solution to the chemotherapy treatment scheduling problem, Theory and Practice of Logic Programming 21 (2021) 835–851.
[42] M. Cardellini, P. D. Nardi, C. Dodaro, G. Galatà, A. Giardini, M. Maratea, I. Porro, A two-phase ASP encoding for solving rehabilitation scheduling, in: S. Moschoyiannis, R. Peñaloza, J. Vanthienen, A. Soylu, D. Roman (Eds.), Proceedings of the 5th International Joint Conference on Rules and Reasoning (RuleML+RR 2021), volume 12851 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 111–125.
[43] M. Alviano, C. Dodaro, M. Maratea, Nurse (re)scheduling via answer set programming, Intelligenza Artificiale 12 (2018) 109–124.
[44] C. Dodaro, G. Galatà, M. Maratea, I. Porro, An ASP-based framework for operating room scheduling, Intelligenza Artificiale 13 (2019) 63–77.
[45] R. Guido, G. Ielpa, D. Conforti, Scheduling outpatient day service operations for rheumatology

diseases, Flexible Services and Manufacturing Journal 32 (2020) 102–128.
[46] M. M. S. El-Kholany, K. Schekotihin, M. Gebser, Decomposition-based job-shop scheduling with constrained clustering, in: J. Cheney, S. Perri (Eds.), Practical Aspects of Declarative Languages, volume 13165 of *LNCS*, Springer, 2022, pp. 165–180.
[47] M. M. S. El-Kholany, M. Gebser, K. Schekotihin, Problem decomposition and multi-shot ASP solving for job-shop scheduling, Theory and Practice of Logic Programming 22 (2022) 623–639. doi:10.1017/S1471068422000217.
[48] G. Francescutto, K. Schekotihin, M. M. S. El-Kholany, Solving a multi-resource partial-ordering flexible variant of the job-shop scheduling problem with hybrid ASP, in: W. Faber, G. Friedrich, M. Gebser, M. Morak (Eds.), Logics in Artificial Intelligence - 17th European Conference, JELIA 2021, Virtual Event, May 17-20, 2021, Proceedings, volume 12678 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 313–328. URL: https://doi.org/10.1007/978-3-030-75775-5_21. doi:10.1007/978-3-030-75775-5_21.
[49] A. Armando, C. Castellini, E. Giunchiglia, M. Idini, M. Maratea, TSAT++: an open platform for satisfiability modulo theories, Electronic Notes in Theoretical Computer Science 125 (2005) 25–36.
[50] L. de Moura, N. Bjørner, Z3: An efficient SMT solver, in: C. R. Ramakrishnan, J. Rehof (Eds.), Tools and Algorithms for the Construction and Analysis of Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 337–340.