Towards an AI planning-based pipeline for the management of multimorbid patients

Malvika Rao[1], Martin Michalowski[2], Szymon Wilk[3], Wojtek Michalowski[1], Amanda Coles[4], and Marc Carrier[5]

[1] University Of Ottawa, Ottawa ON, Canada
mrao@uottawa.ca
[2] University of Minnesota, Minneapolis MN, USA
[3] Poznan University of Technology, Poznan, Poland
[4] King's College London, UK
[5] The Ottawa Hospital, Ottawa ON, Canada

Abstract. Treatment of patients with multimorbidity is one of the greatest challenges for clinical decision support. While evidence-based management of specific diseases is supported by clinical practice guidelines, concurrent application of multiple guidelines requires checking for possible adverse interactions between interventions and mitigating them, before a management plan is constructed. In earlier work, we developed an approach that casts the problem of multimorbidity management as an AI planning problem. In this paper we build on this earlier work and make progress towards creating a pipeline that inputs disease and patient-specific information and outputs a management plan. We describe research focused on selected aspects of pipeline development and illustrate these aspects with a clinical case implemented using the PDDL planning language and the OPTIC planner.

Keywords: multimorbidity · AI planning · end-to-end pipeline

1 Introduction

Clinical practice guidelines (CPGs) and their computer interpretable versions (CIGs) address a single disease whereas patients often suffer from multimorbidity, which is particularly prevalent in older adults [2]. The management of a multimorbid patient requires simultaneous use of multiple CPGs that may recommend disease-specific but overall conflicting treatments resulting in adverse interactions. Thus, the identification and mitigation of adverse interactions and the construction of a management plan, free of these interactions, are crucial components of a clinical decision support [12]. We refer to the identification and mitigation of adverse interactions and the creation of a interaction-free management plan as the *multimorbidity problem*.

In [11] we introduced MitPlan 1.0, that applied AI planning to address the multimorbidity problem. We subsequently expanded our work into MitPlan 2.0 [10], an extended framework that brings our approach fully into the AI

planning paradigm by considering the identification and mitigation of adverse interactions and the creation of a management plan as a single planning problem. This expansion is accomplished using a novel representation that unifies all information pertinent to adverse interactions, enabling the planner to produce the optimal solution (management plan) if one exists. Following typical conventions in solving planning problems, we use the Planning Domain Definition Language (PDDL) to model the problem and a domain independent planner (OPTIC [3]) to solve it.

Our long term goal is to create a pipeline that builds on MitPlan 2.0 and inputs information stored in CIGs, in external sources such as adverse interaction repositories, and in drug ontologies, and provides clinical decision support by generating a management plan with explanations for applied mitigations. Creating such a pipeline requires research on three components: *representation* of clinical information and knowledge that can be reasoned over, *computation* to infer a management plan for complex patient cases, and *explainability* of the inferences made in generating the management plan to help the physician in the treatment of the patient.

In this paper we describe advancements in the representation and computation components of a pipeline. We introduce a new formal representation of revision operators (ROs) – constructs that describe and address adverse interactions – to facilitate the automated translation of input data to PDDL and the detection of adverse interactions. We assume CIGs are given in the form of *actionable graphs* (AGs) that are conceptually based on task-network models [15]. More specifically, AGs are directed graphs with nodes capturing clinical contexts, decisions, actions, and goals, and arcs capturing scheduling constraints between nodes (also allowing for parallel and alternative nodes – see [11] for details).

The computational model is the representation in PDDL of the multimorbidity problem, and we refer to it as *refined computational model* because of the improvements we made to handle complex clinical scenarios and to increase computational efficiency. We describe an automated process that translates AGs and ROs into the refined computational model. This model is then input into the inference engine (in our case the OPTIC planner) to create an internal plan that is subsequently translated into a management plan (with possible explanations). The formalization of ROs and automated translation of AGs facilitate interfacing with external knowledge stored in ontologies and knowledge repositories so this knowledge can be automatically processed and added to the refined computational model.

Illustrative example: To clinically ground and motivate our contributions, we use a simple clinical case as a running example throughout this paper. The case is of a 70 year old male diagnosed with chronic kidney disease (CKD) and hypertension (HTN), and at high risk of developing cardiovascular disease (CVD). Recently this patient has experienced an irregular heartbeat and has been diagnosed with atrial fibrillation (AFib). The patient's prescribed medication includes an erythropoietin stimulating agent (ESA), a calcium channel blocker (CCB), and low dose aspirin.

2 Related work

The multimorbidity problem is an active area of research and several computational approaches have been developed to address it. These approaches are complimentary to the MitPlan 2.0 pipeline in their aim to support the breadth of multimorbidity problem features. Fernandez-Olivares *et al.* [5] describe a temporally-focused multi-agent AI planning approach. In their approach, each agent derives a possible management plan, while patient preferences along with other metrics are used to select the plan that is most suited to the patient. However, their approach does not support some of the clinical complexities such as multiple revisions needed to address adverse interactions. The GLARE-SSCPM system [13] supports physicians to detect and manage adverse interactions as well as merge multiple CIGs. It uses reasoning techniques to address temporal constraints and goals to interactively construct management plan. However there is no optimization involved in generating this plan.

The GoCom [9] system takes a goal-driven approach, where CIG actions are associated with clinical goals and interacting goals are identified and mitigated. Unlike our approach, a physician decides which management plan to select among a number of alternative solutions. Jafarpour *et al.* [8] present a dynamic approach that takes into consideration the evolving nature of a management planning and reconciles different CIGs at execution time. In contrast, MitPlan 2.0 is a static approach designed to be used during a specific patient physician encounter where a treatment time horizon is established. Alaboud and Coles [1] address the problem of managing patients' medication regimes using AI planning to model the continuously changing nature of the multimorbidity problem. However, they focus on medication dosing and do not consider broader mitigation aspects of the problem. Van Woensel *et al.* [14] present a framework where CIGs and evolving patient data are integrated, and adverse interactions mitigated, at execution time according to policies based on clinical knowledge. Their approach supports temporal constraints and employs a local search algorithm to find an optimal task schedule. However, they focus on optimization of an objective function related to temporal constraints and they do not consider other patient or encounter-specific metrics.

3 The MiPlan 2.0 pipeline

In MitPlan 2.0 we leveraged the shared characteristics between the multimorbidity problem and AI planning. In the multimorbidity problem one starts with the current health condition of a patient and the need to establish a management plan. A management plan is a sequence of *clinical actions* (such as administration of medications, tests, etc.) that achieves the *clinical goals* (such as improving health status, addressing a specific complaint, etc.), subject to clinical constraints (such as time, resources, etc.), is free of adverse interactions, and is optimal with respect to selected metrics (such as cost of medication, likelihood of adherence to medication, etc.). Analogously, in AI planning one starts at the

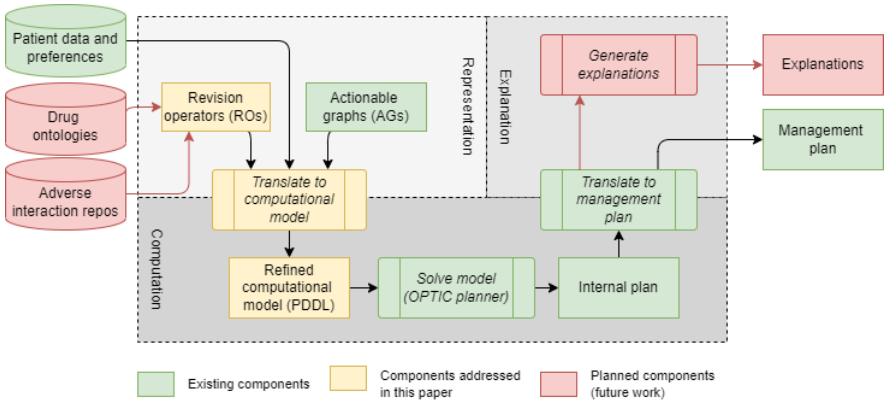

Fig. 1. MitPlan 2.0 pipeline

initial state of a problem and looks for a sequence of *planning actions* from initial to *goal state*, subject to preconditions and effects, such that the identified sequence is optimal with respect to defined metrics.

Clinical actions are activities related to patient treatment as defined in the guidelines and represented in the corresponding AGs as action and decision nodes. These actions are mapped to components of the PDDL planning task as planning objects that are used by planning actions during the process of achieving the goal state. Hence, planning actions operate on planning objects and represent manipulations needed to find a management plan. Relationships between nodes in the AG are captured using predicates and functions. A clinical goal is a node in the AG that, when reached, signifies that the management for the given disease (or part thereof) has been completed and the desired effect has been achieved. The clinical goal is a terminal node in the AG when management planning is to be exhaustive, or it is a node placed somewhere between the context node (root of AG) and a terminal node when a predefined management planning horizon is provided. A *planning goal* is a specification of the goal state to achieve an optimal plan. In our formulation, the planning goal is a conjunction of the goal nodes from all AGs specific to the patient's multimorbid condition.

Information about adverse interactions and ways to mitigate them coming from the guidelines and secondary knowledge sources is encoded in ROs. The MitPlan 2.0 pipeline (see Figure 1) takes CIGs represented as AGs, and ROs as input. Next, ROs and AGs are automatically translated into the PDDL planning problem instance. The computational model consists of the planning problem instance and the planning domain (common for all multimorbidity problems), which defines the actions, predicates and functions needed to solve the planning problem. Technical (programming) advances are introduced into the computational model and the resulting refined computational model is provided to the planner which solves the problem and generates an internal plan, if one exists. The internal plan is then translated into a management plan that is readable

by the physician. As a final step, explanations that clarify what adverse interactions, if any, were identified and how they were mitigated are generated. Thus, MitPlan is customized for a patient/physician encounter. When the patient's health status changes, MitPlan is invoked with the new patient data and a new management plan is created. In this paper we focus on selected components of the pipeline (colored yellow in Figure 1) and describe them in greater detail.

3.1 Revision operators

In this work, we propose a new formal representation of ROs. Each RO mitigates a single clinical adverse interaction and consists of triggering conditions and mitigating actions. Triggering conditions comprise the set of clinical actions and contexts (nodes in AGs) that result in an adverse interaction. Mitigating actions refer to the clinical actions required to mitigate the adverse interaction. Specifically, an RO takes the form of a logical rule: *premise* $\rightarrow$ *conclusion*, where the premise defines the triggering conditions for an adverse interaction and the conclusion (*add*, *replace*) revises parts of an AG by introducing new mitigating actions, or replacing existing clinical actions with new mitigating actions, respectively. For simplicity, MitPlan 2.0 implements the removal of clinical actions as their replacement with a "do nothing" or *no-op* action. Clinical and mitigating actions are associated with metrics such as (but not limited to) *execution cost* (exec), *preference cost* (pref), and the *time duration* (duration) of the action. A predefined default value is assigned if no value is specified for these metrics. A clinical goal does not represent an action. Rather, it just signals that planning should be stopped at this point, and is therefore not associated with any metrics. The planning horizon (and thus the location of a goal node) is decided by the physician for a specific encounter, and clinical goals are not modified by ROs.

Due to space limitations, we describe only the *replace* operation. The reserved keyword *replace* is followed by the set of clinical actions that must be replaced and then by the set of mitigating actions that replace them. We use the special terms (*or*, *and*, *then*) to indicate when the set of mitigating actions are alternatives to one another, must be carried out in parallel, or must be carried out in sequence, respectively. Specific examples are described below.

In MitPlan 2.0, the objective function minimizes a weighted sum of user-identified metrics, where the weights are chosen to reflect the metrics' relative importance, and both metrics and weights are tailored to a specific clinical context. Patient preferences are captured by alternative actions with different preference costs, such that an alternative action preferred by a patient is set to be less costly. Information about patient preferences is acquired and either added directly to the AG by associating alternative actions with preference costs, or included as part of ROs where newly introduced actions are associated with costs reflective of those preferences.

By default, we assign an execution cost to each planning object in the computational model, and the objective function minimizes overall cost (including preference cost). Clinical actions introduced by ROs to mitigate adverse interactions have a higher execution cost than the clinical actions they revise. As

a result the planner always attempts to select the original clinical actions. If a feasible plan does not exist because of adverse interactions, more costly actions specified in the ROs are selected. If several actions are equally suitable to mitigate an adverse interaction, the planner selects those actions that minimize the value of the objective function (alternatively there might exist multiple optimal solutions).

Illustrative example (cont.): We show how the ROs are represented in the clinical case introduced in Section 1. This case involves adverse interactions which are mitigated with a set of three ROs (RO_1, RO_2, RO_3), described next. For illustrative purposes we use a default execution cost of 100.

- RO_1 : (AFib, CCB, BB) $\rightarrow$ {*replace*(BB, no-op[exec: 100])}
  For a patient diagnosed with AFib, the simultaneous clinical actions of prescribing CCB and BB (beta blocker) medications represent an adverse interaction (triggering condition). The clinical action of prescribing BB is removed (mitigating action).
- RO_2 : (CKD, PCB) $\rightarrow$ {*replace*(PCB, SCB[exec: 100])}
  A patient diagnosed with AFib might be prescribed a potassium channel blocker (PCB) for anti-arrhythmic therapy. The triggering condition specifies that PCB prescribed to a patient diagnosed with CKD represents an adverse interaction. The clinical action of prescribing PCB is replaced with the mitigating action of prescribing a sodium channel blocker (SCB) medication.
- RO_3 : (AFib, low-dose-aspirin) $\rightarrow$ {*replace*(low-dose-aspirin, warfarin[exec: 100; pref: 20] *or* DOAC[exec: 100; pref: 5])}
  For a patient diagnosed with AFib, prescribing low dose aspirin (for CVD prevention) represents an adverse interaction (triggering condition). The clinical action of prescribing low dose aspirin is replaced with the mitigating action of prescribing an anticoagulant such as warfarin or a direct oral anticoagulant (DOAC). According to this RO, the patient's preference is to be prescribed DOAC rather than warfarin as seen in the preference cost of the action associated with prescribing DOAC being lower than that of the action prescribing warfarin.

Multi-action revisions: Building further on the illustrative example, we implement multi-action revisions, that is, a situation where the conclusion component of an RO consists of a sequence of mitigating actions rather than a single action. Treating CKD and HTN requires lifestyle management as part of the patient's treatment and lifestyle management may vary from patient to patient, as illustrated below.

- RO_4 : (CKD, HTN, lifestyle-management) $\rightarrow$ {*replace*(lifestyle-management, DASH-diet[exec: 100] *then* sodium-intake-restriction[exec: 100])}

The triggering condition indicates that, for a patient diagnosed with CKD and HTN, and told to manage their lifestyle in a non-specific way, their generic lifestyle management is replaced by two mitigating actions described in the conclusion of RO_4 – the DASH diet followed by restricting sodium intake.

Temporal constraints: In the illustrative example, a patient's CVD risk is managed with a DOAC as anticoagulation therapy. Considering that CKD patients are predisposed to oral lesions and tooth decay, it is determined that the patient needs to undergo a dental procedure that is associated with a high risk of periprocedural bleeding. The patient's anticoagulation medication (DOAC) needs to be stopped 2 days prior to the procedure and restarted 1 day after the procedure. This requirement is represented using a sequence of actions in the conclusion of the revision operator RO_5.

– RO_5 : (CKD, AFib, DOAC, dental-procedure) $\rightarrow$ {*replace*((DOAC, no-op[exec: 100; duration: 4 days] *then* DOAC[exec:10; duration: lifetime])}

3.2 Translation to computational model

MitPlan 2.0 creates the computational model (*Translate to computational model* in Figure 1) with a Python module using the NetworkX library to manipulate the AGs extended with the ROs, patient data and preferences. The manipulated AGs are automatically translated into a unified representation within the PDDL planning problem instance for a given patient encounter [4]. This unified problem representation enables the planner to optimize over all the information available, including contingencies introduced by ROs and patient preferences, and it is constructed in two steps: (1) the AG is expanded to include all mitigating actions recommended by applicable ROs as well as alternative actions required for representing patient preferences, and (2) the triggering condition of each RO is encoded as a binary vector identifying the set of clinical actions and contexts that define an adverse interaction.

Illustrative example (cont.): During translation, the unified internal problem representation for the clinical case involving the first three ROs is constructed as follows. No-op actions are introduced to the AG for AFib, with a higher cost than that of BB medication, as alternative actions to those representing BB medication, to model the removal of BB medication as per RO_1. In addition, the AG for AFib is expanded to include an action for SCB, as per RO_2. As per RO_3, the AG for CKD is expanded to include two alternative actions – an action for prescribing a DOAC and an action for prescribing warfarin, with a lower preference cost associated with DOAC than warfarin to represent the patient's preference for the former over the latter. Finally, the triggering conditions for RO_1, RO_2, and RO_3 are encoded as binary vectors.

3.3 Refined computational model

A recent review of the literature on multimorbid disease management identified a set of key features that characterize a multimorbid problem [12]. Implementing some of these features, such as delaying a treatment to avoid an overlap or capturing a temporal relationship between AGs, required improving the representation of the planning problem. As a result, we were able to implement more complex mitigations, consider AGs with a large number of nodes, and deal with sequences of parallel clinical actions including nested parallel actions. We handled the processing of parallel actions, and incorporated temporal constraints by using durative actions in PDDL to associate a time duration with an action. We were also able to handle situations where the conclusion component of an RO involves a set of mitigating actions.

All these additions resulted in a significant increase in the planner's search space and consequently growth in its runtime. In order to address this issue we revised the problem representation to reduce the planning search space. For example, we introduced preconditions in planning actions to direct the planner's search away from taking certain actions, where those actions might lead to an interaction or prevent reaching the goal. We also modelled certain planning actions as PDDL+ [7] events which execute instantly when their preconditions hold. This reduced the number of possible choices the planner considered and sped up search. Incorporating these changes and solving the refined computational model required the use of the versions PDDL 2.1 [6], PDDL+, and the OPTIC planner which supports it.

Illustrative example (cont.): As a result of the additions and refinements described above, we were able to generate an internal plan within reasonable processing time. Figure 2 shows such a plan for the illustrative example introduced earlier. In this plan we have highlighted (in bold) an action of prescribing SCB (labeled as *newscb*) that replaces an action of prescribing PCB as defined by RO_2.

4 Conclusion and future work

In this paper, we presented selected improvements that move us closer to completing the MitPlan 2.0 pipeline. We formalized the representation of the ROs to facilitate their creation using external repositories. We described how the AGs, ROs, patient data and patient preferences are automatically translated into a computational model expressed in PDDL. We showed how the computational model was refined in order to reduce the search space and support complex multimorbidity problems. All of these advancements enable MitPlan 2.0 to solve larger and more complex clinical cases and therefore position our work closer to prospective evaluation.

Our future work will focus on generating explanations for specific mitigations, moving towards creating ROs from drug ontologies and adverse reaction

0.023: (makedecisionnode d afibtype cardio) [0.010]
0.034: (makedecisionnode d3 bpcontrol2 p1) [0.010]
0.034: (makedecisionnode d2 ferritin metabolicabnormality) [0.010]
0.034: (takeactionnode d cardio improve) [0.010]
0.045: (makedecisionnode d2 metabolicabnormality p5) [0.010]
0.045: (makedecisionnode d improve recur) [0.010]
0.045: (executeparallelstartnode d3 p1) [0.010]
0.056: (**makedecisionnode d recur newscb**) [0.010]
0.056: (executeparallelstartnode d2 p5) [0.010]
0.056: (takeparallelactionnorevisions d3 p1 pace12wks pace12wks_end) [0.010]
0.056: (takeparallelactionnorevisions d3 p1 pdiuret12wks pdi12wks_end) [0.010]
0.067: (**takeactionnode d newscb g**) [0.010]

.

.

.

0.149: (checkgoal d g) [0.001]
0.149: (checkgoal d2 g2) [0.001]
0.149: (checkgoalnorevisionops d3 g3) [0.001]

Fig. 2. Part of the internal plan for the illustrative example

repositories, and enriching the semantic representation of clinical actions to support reasoning at the level of medication classes and specific medications within them. An interesting question raised by this research is the notion of clinical case complexity. The number of AGs, their size and structure, the number and types of interactions, the constraints on the problem, and the size of the search space seem to be factors that contribute to the complexity of a clinical case, but their interplay is uncertain. Therefore, our future research will also explore the notion of clinical case complexity and what factors impact solvability of the multimorbidity problem.

5 Acknowledgements

We thank Jean-Luc Blais-Amyot and Maxime Côté-Gagné for their programming work on the automated translation component. We thank the reviewers for their helpful feedback. This research was supported by funding from the Telfer Health Transformation Exchange and the Natural Sciences and Engineering Research Council of Canada.

References

1. Alaboud, F.K., Coles, A.: Personalized medication and activity planning in pddl+. In: Proceedings of 29nd International Conference on Automated Planning and Scheduling (ICAPS). pp. 492–500 (2019)
2. Barnett, K., Mercer, S., Norbury, M., Watt, G., Wyke, S., Guthrie, B.: Epidemiology of multimorbidity and implications for health care, research, and medical education: a cross-sectional study. The Lancet **380**, 37–43 (2012)

3. Benton, J., Coles, A.J., Coles, A.: Temporal planning with preferences and time-dependent continuous costs. In: Proceedings of 22nd International Conference on Automated Planning and Scheduling (ICAPS). vol. 22, pp. 2–10. AAAI Publications (2012)
4. Blais-Amyot, J.L., Cote-Gagne, M.: Mitplan generation of pddl based on cpgs. Honours project report, University of Ottawa (2021)
5. Fernandez-Olivares, J., Onaindia, E., Castillo, L., Jordan, J., Cozar, J.: Personalized conciliation of clinical guidelines for comorbid patients through multi-agent planning. Artificial Intelligence in Medicine **96**, 167–186 (2019)
6. Fox, M., Long, D.: PDDL2.1: An extension to PDDL for expressing temporal planning domains. Journal of artificial intelligence research **20**, 61–124 (2003). https://doi.org/10.1613/jair.1129
7. Fox, M., Long, D.: Modelling mixed discrete-continuous domains for planning. Journal of artificial intelligence research **27**, 235–297 (2006)
8. Jafarpour, B., Raza, S., Van Woensel, W., Sibte, S., Abidi, R.: Execution-time integration of clinical practice guidelines to provide decision support for comorbid conditions. Artificial Intelligence in Medicine **94**, 117–137 (2019)
9. Kogan, A., Peleg, M., Tu, S.W., Allon, R., Khaitov, N., Hochberg, I.: Towards a goal-oriented methodology for clinical-guideline-based management recommendations for patients with multimorbidity: Gocom and its preliminary evaluation. Journal of Biomedical Informatics **112** (2020)
10. Michalowski, M., Rao, M., Wilk, S., Michalowski, W., Carrier, M.: Mitplan 2.0: Enhanced support for multi-morbid patient management using planning. In: Tucker, A., Abreu, P.H., Cardoso, J., Rodrigues, P.P., Riano, D. (eds.) Artificial Intelligence in Medicine. 19th Conference on Artificial Intelligence in Medicine, AIME 2021, Porto, Portugal, June 15 – June 18, 2021. Proceedings, pp. 276–286. Springer (2021)
11. Michalowski, M., Wilk, S., Michalowski, W., Carrier, M.: A planning approach to mitigating concurrently applied clinical practice guidelines. Artificial Intelligence in Medicine **112** (2021)
12. O'Sullivan, D., Van Woensel, W., Wilk, S., Tu, S.W., Michalowski, W., Abidi, S., Carrier, M., Edry, R., Hochberg, I., Kingwell, S., Michalowski, M., O'Sullivan, H., Peleg, M.: Towards a framework for comparing functionalities of multimorbidity clinical decision support: A literature-based feature set and benchmark cases. To appear in AMIA 2021 (2021)
13. Piovesan, L., Terenziani, P., Molino, G.: Glare-sscpm: An intelligent system to support the treatment of comorbid patients. IEEE Intelligent Systems **33**(6), 37Ñ46 (2018)
14. Van Woensel, W., Abidi, S., Abidi, S.: Decision support for comorbid conditions via execution-time integration of clinical guidelines using transaction-based semantics and temporal planning. Artificial Intelligence In Medicine **118** (2021)
15. Wilk, S., Michalowski, M., Michalowski, W., Rosu, D., Carrier, M., Kezadri-Hamiaz, M.: Comprehensive mitigation framework for concurrent application of multiple clinical practice guidelines. Journal of Biomedical Informatics **66**, 52–71 (2017)