Mechatronics 34 (2016) 38–62

Contents lists available at ScienceDirect

Mechatronics

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

Symbolic discrete-time planning with continuous numeric action parameters for agent-controlled processes

Florian Pantke *, Stefan Edelkamp, Otthein Herzog

Center for Computing and Communication Technologies (TZI), University of Bremen, Am Fallturm 1, 28359, Germany

ARTICLE INFO

Article history:
Available online 3 August 2015

Keywords:
Artificial intelligence planning
Non-linear optimization
Infinite-domain constraint satisfaction
Interval analysis
Multi-agent system
Manufacturing control

ABSTRACT

In industrial domains such as manufacturing control, a trend away from centralized planning and scheduling towards more flexible distributed agent-based approaches could be observed over recent years. To be of practical relevance, the local control mechanisms of the autonomous agents must be able to dependably adhere and dynamically adjust to complex numeric goal systems like business key performance indicators in an economically beneficial way. However, planning with numeric state variables and objectives still poses a challenging task within the field of artificial intelligence (AI).

In this article, a new general-purpose AI planning approach is presented that operates in two stages and extends existing domain-independent modeling formalisms like PDDL with continuous (i.e., infinite-domain) numeric action parameters, which are currently still unsupported by state-of-the-art AI planners. In doing so, it enables the solution of mathematical optimization problems at the action level of the planning tasks, which are inherent to many real-world control problems. To deal with certain difficulties concerning reliable and fast detection of action applicability that arise when planning with real-valued action parameters, the implemented planner allows resorting to an adjustable "satisficing" strategy by means of partial execution and subsequent repair of infeasible plans over the course of time. The functioning of the system is evaluated in a multi-agent simulation of a shop floor control scenario with focus on the effects the possible problem cases and different degrees of satisficing have on attained plan quality and total planning time. As the results demonstrate the basic practicability of the approach for the given setting, this contribution constitutes an important step towards the effective and dependable integration of complex numeric goal systems and non-linear multi-criteria optimization tasks into autonomous agent-controlled industrial processes in a reusable, domain-independent way.

1. Introduction

In production planning and control, a shift of attention from centralized control and scheduling to novel distributed multi-agent based approaches could be observed over recent years, as conventional systems more frequently cannot satisfy the increased need for flexibility within the respective processes that is caused by a continuing trend towards higher degrees of product customization, decreasing lot counts, shorter product lifecycles, and preference of just-in-time production and delivery. To stay competitive, organizations must tackle production planning and control tasks, which have significantly grown in structural and dynamic complexity as a result of these changes [1,2]. When central control becomes impractical due to its computational cost and lack of reactivity to rapidly changing environmental influences, farming out the solution process to several interacting software agents appears to be an appropriate method to meet the new requirements [3–7].

Such agents are faced with sets of various qualitative and quantitative business objectives, e.g., given in terms of key performance indicators (KPIs) of the application domain, often with some of them mutually conflicting. To be of industrial relevance, an agent-based control approach must therefore be able to dependably adhere and dynamically adjust to complex numeric goal systems in an economically beneficial way. In this respect, mathematical optimization is a common device for the identification of acceptable trade-offs between multiple criteria [8–10]. With communication protocols and approaches for inter-agent coordination based on user-defined KPIs having been proposed [11], the agents' local control components must be able to properly model and solve mixed combinatorial-numeric planning and optimization problems. However, planning and scheduling with numeric goal systems at present still poses a challenging task

* Corresponding author. Tel.: +49 421 218 64000; fax: +49 421 218 64047.
E-mail address: fpantke@tzi.de (F. Pantke).

http://dx.doi.org/10.1016/j.mechatronics.2015.06.015
0957-4158/

within the field of computational intelligence. It is a well-known result that already the most basic classes of numeric planning problems are, in general, undecidable with respect to plan existence [12]. As a consequence of this, in practice, the execution of slightly suboptimal, defective, or incomplete plans, possibly combined with an incremental improvement by means of subsequent re-planning, may often be preferable to arriving at no solution at all within the available timeframe. General-purpose artificial intelligence (AI) planners and suitable application-independent modeling formalisms that support non-linear mathematical optimization problems at the action-executing level are still quite rare due to the difficulty (e.g., NP-hardness) of this class of planning tasks. While the Planning Domain Definition Language (PDDL), which is widely used for modeling in AI planning, offers numeric state variables (termed "fluents" in PDDL) and continuous numeric change over time [13], none of its official versions allows for continuous numeric action parameters, which are required for certain real-world planning and optimization problems, such as can be found in industrial domains like manufacturing control (e.g., different manufacturing parameters of machine tools, optimization of resource usage, etc.).

1.1. Contribution and outline of this article

In this article, an AI planning system for local agent control is presented that adds real-valued parameters to its discrete-time action model and couples combinatorial planning with numeric optimization in a two-stage process, which utilizes interval analysis [14–17] and fast constraint consistency techniques [18] for action applicability tests. To further facilitate the modeling of KPI-based goal systems and thereby increase its practical usefulness, the planner supports the automatic aggregation of user-defined key figures over dynamic groups of planning objects and the proactive modification of these groups with the planned and executed actions. Due to the lifting of the intermediate plans to an interval extension, proper determination of the real-valued satisfiability of their combined set of numeric action conditions becomes computationally difficult. In the combinatorially constructed plans, conflicts between numeric constraints may exist and, therefore, have to be recognized and repaired. This paper gives a detailed look into the nature of several of such problematic situations with respect to constraint satisfiability and action applicability that might occur in practice and proposes a parameterized "satisficing" [19] strategy for handling them in the planning process. The presented planning approach employs partial plan execution and dynamic re-planning over time for addressing resultant plan defects and may vary in success depending on the given application domain. The operative performance of the approach is examined in a proof-of-concept case study with a simulated shop floor, on which incoming customer orders need to be assigned to different machine tools, each one managed by an autonomous agent. As the results show the basic practicability of the approach in the given scenario, this contribution constitutes an important step towards the effective and dependable integration of complex numeric goal systems and non-linear multi-criteria optimization tasks into autonomous agent-controlled industrial processes and cyber-physical systems in a reusable, domain-independent way.

The present article is an extended and revised version of a research paper [20] published by the authors at the 2nd International Conference on System-Integrated Intelligence (SysInt 2014). It amends the conference publication with additional experimental data and a more detailed description of the two-stage planning algorithm and its symbolic planning model. The article is structured as follows. First, the multi-agent shop floor scenario, which serves as a running example throughout the text, is presented in Section 2, and an introduction to the basic modeling of AI planning tasks is given. Section 3 starts with an explanation of why continuous action parameters are so difficult to handle and how they are used in the planning domain of this article. Following this, the methodology of the proposed two-stage approach to numeric planning with continuous parameters is described in depth, using several problem cases taken from the evaluation scenario as well as some minimal examples for illustration purposes. The focus here lies on the core problem of correctly determining the satisfiability of the numeric action conditions in a given plan, which is undecidable in the worst case. In Section 4, the domain-independent planning algorithm is summarized in pseudocode, and the mathematical optimization problem of choosing the final real-valued action parameter assignments is formally specified. Section 5 discusses the aspect of automatic KPI aggregation during planning and exemplifies how the well-known dependency problem of interval arithmetic may impede numeric planning performance. In Section 6, some domain-specific pruning and control rules, which are applied in the experiments to reduce the size of the planning search space, are given. The behavior of the domain-specific plan enumerator, which results from these rules, is described in pseudocode. After this, the obtained experimental results are presented and examined particularly with regard to the influence the chosen degree of satisficing has on attained plan quality. Also, the performance of the proposed approach is compared to conventional PDDL-based planning with discrete action parameters. Section 7 discusses related work in the field of AI planning. The article closes with a short conclusion and outlook in Section 8.

2. Modeling AI planning for shop floor control

For the evaluation of the proposed planning algorithm, the application domain of the IntaPS [21] multi-agent system for the simulation of integrated process planning and production control was chosen. This software system has been developed over several years at the Center for Computing and Communication Technologies (TZI) at the University of Bremen in collaboration with the Institute of Production Engineering and Machine Tools (IFW) at the University of Hanover in a forerunner project to and in the context of the Collaborative Research Center (CRC) 637 "Autonomous Cooperating Logistic Processes" [22]. The main objective of the IntaPS approach is to shift decision-making tasks from centralized, and possibly separate, process planning, production scheduling, and production control processes towards an integrated process planning and control mechanism for make-to-order production and mass customization by means of distributed, self-monitoring autonomous agents and associated communication protocols. While previous research [21,23,24] has dealt with the question of how process planning, e.g., with respect to process chains over multiple machines, can efficiently be performed on-the-fly together with actual machine allocation during production scheduling, the present article focuses on the combination of autonomous planning and automation control, and in particular the numeric optimization of the machining parameters in the planning and scheduling task.

2.1. Management of manufacturing orders in the IntaPS approach

The IntaPS scenario, the order-taking procedure differs from settings where a human decision maker manually accepts incoming manufacturing requests from external customers, e.g., in a first-in, first-out (FIFO) fashion and based on an estimation that takes the current order backlog, machine utilization, stock, and lead times into account. In such approaches, the time-consuming scheduling of the accepted set of orders is done at a later time after