Rough Scheduling: A Research Problem

Julia Johnson
Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2, E-mail: johnson@cs.uregina.ca

Abstract: *Rough mereology (Polkowski and Skowron., 1996) is a rough sets-based approach to problems of approximate reasoning extended so that the problems can be allocated to a system of agents communicating with each other in natural language. It comprises a process of learning about how objects are constructed and a process of synthesizing objects based on the knowledge acquired in the learning process. Whereas the rough mereology has previously been applied to control problems (to provide a rough mereological controller), I am proposing to apply it to production scheduling in manufacturing (to develop a rough mereological scheduler). This problem provides a framework to investigate a number of specific problems in optimization, natural language processing, knowledge representation and learning.*

1 Introduction

A production schedule is a week by week plan showing which products are produced on which processors in which time periods. A feasible schedule satisfies demands. An optimal schedule minimizes total production cost while satisfying demands.

A new scheduling method is proposed differing from optimization techniques [1, 2, 6, 8, 19] and constraint programming [3, 20, 21, 23]. Close to optimal schedules are produced which satisfy their requirements "in acceptable degree".

1.1 Rough Scheduling

Rough mereology [22] offers a new paradigm for production scheduling because a solution is expressed in approximate terms, that is 'in acceptable degree', unlike previous approaches in which an optimal solution was expected. Decomposition formulae are extracted from examples of near-feasible schedule constructions from near-feasible regions. A synthesis process combines near-feasible regions to produce a near-feasible schedule. We assume that acceptability of the cost of a schedule is independent from acceptability of customer demands. The cost of an acceptable schedule being constructed is not incremented beyond the cost of the example schedules because the example schedules have tighter constraints than the schedule under construction.

1.2 Rough equivalence of NL expressions

Ambiguous language and equivalence of statements in language in terms of their meaning are problems in natural language (NL) processing. Since equivalence is undecidable, rough sets methods should provide a metric to judge when two language statements are "similar enough" in meaning. Within the agent based setting, my interest is in how to tell whether the expression of the object in the internal language of the agents satisfies the expression of the external requirement for the object.

1.3 A language for internal representation

REFINE is an extension of DEFINE (a set theory database language; Gilmore, 1987) to deal with domains characterized by incomplete and insufficient information. Our objective is to develop REFINE for internal representation of NL messages and to use it to assist in implementing an agent based system.

1.4 Factors that a system does not know

It would be useful to recognise when a system is unable to provide the association between a value (an instance of a concept) and its attribute (a representation of that concept). If this can be recognized, then a resolution is possible, e.g., the system could report on its lack of knowledge. Recognizing factors that an agent does not know is important for systems of autonomous agents communicating with each other in natural language.

1.5 Integration

Successful integration of these parts will provide a solution to an industrial problem in support of Canadian efforts to make its manufacturing sector more competitive.

2 Previous Work

2.1 Scheduling. [Johnson & Chen]

In [13], we gained experience with the production scheduling domain. An object-oriented approach to the optimization of production schedules was presented. Expert system software was developed to modify the schedule in a post-processing step to include scheduler preferences. That system could not guarantee that the cost of the schedule would not be incremented beyond an acceptable level.

In [10], Dynamic Set-Relation net (DSRN) is applied to production scheduling in manufacturing. The dynamic properties of DSRN permit production, demand and inventory to be modeled in terms of transitions and firing rules. The Petri net (specifically Relation net) modeling is formulated in the context of the design of decision support systems for manufacturing. In [7], the Petri net representation is incorporated along with discrete events simulation to simulate the scheduling of different types of job processes in flexible manufacturing systems. The SIMSCRIPT simulation language was used to find, for each job, the state of each workstation in all possible sequences and to select an optimal sequence for doing the jobs.

2.2 Rough equivalence of natural language expressions. [Johnson & Rosenberg]

In [17], we provide a heuristic based on the notion of semantic relatedness for determining the order in which paraphrases for reasonable interpretations for natural language requests should be presented to the user (most likely first). Results in resolving ambiguous language and schema equivalence are reported in [14]. We introduce rough sets to SET schemas in [15].

2.3 A language for internal representation [Johnson & Liu]

In [16], we used semantic knowledge from the meta-database to address database issues and natural language processing simultaneously suggesting that a similar overlap could be used to advantage for translating human generated texts to goals for a machine-agent. In [12], we introduced the notion of Rough Set to the SET Conceptual model to develop the database language REFINE to deal with domains characterized by incomplete and inconsistent information. (Same name but different system from Reasoning System's software re-engineering system, Software Refinery and REFINE/C.)

2.4 Identifying factors that a system does not know. [Johnson & Rosenberg]

The objective is to recognize when a system is unable to provide the association between a value (an instance of a concept) and its attribute (a representation of that concept). If this can be recognized, then a resolution is possible, e.g., the system could report on its lack of knowledge. In [18] this problem is avoided (not resolved) by integrating the lexicon and the database. Our problem differs from current work on cooperative answering addressed by means of presuppositions [26]. The basic assumption underlying that work is that the user's mental model needs to be corrected by the system. The basic assumption underlying our work is that the system's mental model needs to be corrected to conform to that of the user.

2.5 Integration

In [9], we consider the problem of agents communicating with each other by passing messages expressed in natural language. The agents are cooperating to achieve a common goal. Internal representation of the natural language messages occurs in REFINE which includes rough set notations directly in the language. Such internal representation facilitates knowledge acquisition by the agents.

3 Solution Plan

3.1 Rough Scheduling

Long term objective is to produce close to optimal schedules which satisfy their requirements "in acceptable degree".
Review & Methods: The *rough mereology* approach to synthesis of complex objects will be used to produce sub-optimal schedules that are acceptable. Acceptability criteria is defined by the human scheduler and by the customers. Requirements imposed by the human scheduler (e.g., product sequencing constraints) must be satisfied "in acceptable degree". Similarly, customer demands are to be satisfied "in acceptable degree".

Rough mereology (RM) has previously been formulated within the context of control problems (to provide a rough mereological controller [25]). I will apply rough mereology to production scheduling in manufacturing (to develop a rough mereological scheduler). Input to the RM scheduler is a function that is learned from examples of the construction of acceptable schedules. Relaxing the feasibility requirement (customer demands are satisfied in acceptable degree) tends to decrease the cost of the schedule. Likewise, relaxing scheduler preferences tends to decrease the cost of the schedule. Hence, the cost of an RM schedule is bounded by the cost of the example schedules. In this sense, the cost of a schedule is not incremented beyond an acceptable level.

Adapting the approach of [22, 25], we construct object synthesis schemes that describe how objects are constructed from sub-objects. The rough inclusion function measures similarity between objects by the proportion of attributes that the objects have in common. Other similarity measures are possible. The one presented in [14] has the property that the outcome is invariant to arbitrary decisions made by the person who designed the information tables. The rough inclusion function does not have this property. Different measures will be obtained depending on the numbers and types of attributes for the objects.

We are initiating research into integrating human and machine agents. In [24], all of the agents are machine–agents. However, given the practice of including scheduler preferences in a post processing step, a better design for our problem would be to distinguish one of the agents as a human agent. In this regard, our approach is similar to that of [4] where it is argued that human dialogue behaviors differ from machine agent interactions.

There are competing scheduling objectives: For example, feasible schedule construction and the inclusion of scheduler preferences are competing goals. For illustration of the latter, consider a human scheduler who will prefer to run widgets before gadgets on a single processor if widgets are white and gadgets are black. Otherwise, the processor will have to be shut down for cleaning during a switch from gadgets to widgets. Feasible schedule construction occurs by means of machine agent–machine agent interactions while scheduler preferences are introduced by means of human agent–machine agent interactions. Hence the production scheduling problem allows both paradigms to be investigated.

Our *applied research* will provide a solution to the development of intelligent software systems for industrial and business problems. Our *theoretical research* is relevant to object synthesis in general. Continuing activities include: 1) showing how general constraints of the rough mereology force a proper design for partitioning a problem into subproblems for allocation to a hierarchy of agents. 2) formulating an infeasible solution to an MP model as a schedule that satisfies customer demands in acceptable degree while keeping cost of the schedule at an acceptable level.

3.2 Rough equivalence of NL expressions

Short term objective is to develop a measure of interestingness of the human input in terms of machine agent goals.
Review & Methods: The object described by the human agent is understood by the machine-agents only in approximate terms. We will investigate the question of which parts of the meaning of the input should be included. In [4] for appointment scheduling, a human agent analyses human generated texts transforming interesting parts of their meaning to statements accepted by a machine agent. Adapted to production scheduling, our system analyses human generated texts that describe desirable schedules, transforming them to expressions that easily map onto machine–agent primitives. An example of a human input is "the white widgets are run before the black gadgets on processor P". Previous measures of interestingness have been based on "informativeness" and "contextual relevance"[4]. For appointment scheduling, temporal information in the input is especially interesting. For production scheduling, information about cost and color of the products would be more interesting. In [13], we mapped concepts that are meaningful for scheduling (products, processors, time periods, product numbers, idle time, busy time) to primitives for 2-D robot movements (blocks, horizontal grid position, vertical grid position, block color, empty grid position, non-empty grid position, respectively). However, task specific information was encoded in the Prolog programs that implemented the system. In contrast, the rough mereology approach allows different agents to view the same object from different perspectives avoiding the need to handcode mappings between different agent views. However, this greater generality will require the use of a descriptive language to support inference of a strategy for interaction among agents similar to [5] but requiring an acceptable rather than optimal solution.

3.3 A language for internal representation

Short term objective is to translate interesting parts of NL texts to goals for machine-agents.
Review & Methods: Rough set notations have been included directly in a query language REFINE [11, 12] to permit part of the meaning to be expressed. REFINE supports concepts not explicitly represented by data. This ability facilitates reasoning about queries for managing the translation from human-agent language to machine-agent language. It remains to include in REFINE approximations of concepts to provide a language particularly suited to integrating human and machine agents.

4 Conclusion

Production schedules can be produced more efficiently within an approach to reasoning under uncertainty because the schedules do not have to be optimal. We produce, by means of the rough mereology, suboptimal schedules that are acceptable. Rough mereology applied to scheduling illustrates an approach for finding acceptable solutions to NP-Hard problems in general.

References

[1] S.G. Abraham, V. Kathail, and B.L. Beitrich. Meld scheduling: A technique for relaxing

scheduling constraints. *International Journal of Parallel Programming*, 26(4):349–381, 1998.

[2] M.C. Agar and S. Salhi. Lagrangean heuristics applied to a variety of large capacitated plant location problems. *Journal of the Operational Research Society*, 49(10):1072–1084, 1998.

[3] C. Aravindan, J. Dix, and I. Niemela. A research project on disjunctive logic programming. *AI Communications*, 10:151–165, 1997.

[4] S. Buseman, T. Declerck, A.K. Diagne, L. Dini, J. Klein, and S. Schmeier. Natural language dialogue service for appointment scheduling agents. In *Proceedings of the 5^{th} Conference on Applied Natural Language Processing*, pages 25–32, 1997.

[5] D. Carmel and S. Markovitch. Model-based learning of interaction strategies in multi-agent systems. *Journal of Experimental and Theoretical AI*, 10(3):309–332, 1998.

[6] B. Chen, C.N. Potts, and V.A. Strusevich. Approximation algorithms for two–machine flow shop scheduling with batch set up times. *Mathematical Programming*, 82:255–271, 1998.

[7] L. Chen and J.A. Johnson. Relation nets and database query. In *Proceedings of the $3^r d$ Pacific Rim International Conference on Artificial Intelligence*, pages 752–758, Beijing, China, 1994.

[8] L.A. Hall. Approximability of flow shop scheduling. *Math. Programming*, 82:175–190, 1998.

[9] J. Johnson. Rough mereology for industrial design. In L. Polkowski and A. Skowron, editors, *Rough Sets and Current Trends in Computing*, pages 553–556. Springer, 1998.

[10] J. Johnson and L. Chen. Relation nets for production scheduling in manufacturing. In G. Rzevski, P.A. Adey, and D.W. Russell, editors, *Applications of AI in Engineering IX*, pages 586–592. Computational Mechanics, 1994.

[11] J. Johnson and M. Liu. The language REFINE. In *Proc. 4^{th} Joint Conference in Information Science JCIS'98*, volume II, pages 367–370, 1998.

[12] J. Johnson and M. Liu. Rough sets for informative question answering. In *Proc. ICCI'98 9^{th} International Conference on Computing and Information*, pages 53–60, 1998.

[13] J.A. Johnson. Optimization without computation: An application of artificial intelligence to product scheduling in manufacturing. In *Fourth National Conference on Robotics*, volume 1, pages 235–242, 1993.

[14] J.A. Johnson. Semantic relatedness. *Computers & Math. with Applications*, 29(5):51–64, 1995.

[15] J.A. Johnson. Rough set schemas. In *Proc. 5^{th} International Conference Information Systems Development - ISD'96*, pages 193–203, 1996.

[16] J.A. Johnson and R.S. Rosenberg. Knowledge sharing for natural language understanding and conceptual modeling for database design. In *Proc. PRICAI'92*, pages 980–988, 1992.

[17] J.A. Johnson and R.S. Rosenberg. A measure of semantic relatedness for resolving ambiguities in natural language database requests. *Data & Knowledge Engineering*, 7(3):201–225, 1992.

[18] J.A. Johnson and R.S. Rosenberg. Using the database as part of the lexicon: Harris' idea revisited. In *Proceedings of the 3^{rd} Pacific Rim International Conference on Artificial Intelligence*, pages 635–641, Beijing, China, 1994.

[19] O.-H. Kwon and K.-Y. Chwa. An algorithm for scheduling jobs in hypercube systems. *IEEE Transactions on Parallel and Distributed Systems*, 9(9):856–860, 1998.

[20] K. Marriott and P. Stuckey, editors. *J. Logic Programming: Special Issue on Constraint Logic Programming*, volume 37. North-Holland, 1998.

[21] K. Marriott and P. Stuckey. *Programming with Constraints: An Introduction*. MIT Press, Cambridge, Mass., 1998.

[22] Lech Polkowski and Andrzej Skowron. Rough mereology: A new paradigm for approximate reasoning. *Int. Journal of Approximate Reasoning*, 15(4):333–365, 1996.

[23] V.A. Saraswat, R. Jagadeesan, and V. Gupta. Default timed concurrent constraint programming. In *Proceedings POPL'95*, 1995.

[24] Andrzej Skowron and Lech Polkowski. Rough mereology and analytical morphology. In E. Orlowska, editor, *Incomplete Information: Rough Set Analysis*, chapter 13, pages 399–437. Physica Verlag, 1998.

[25] Andrzej Skowron and Lech Polkowski. Rough mereology foundations for design, analysis, synthesis, and control in distributed systems. *Information Sciences*, 104(1-2), 1998.

[26] P.G. Surette and R.E. Mercer. Towards defining presuppositions in a Montague grammar-like framework. In *Proceedings Pacific Association for Computaional Linguists*, pages 335–349, 1999.