Energy Efficient Production Planning

A Joint Cognitive Systems Approach

Connor Upton[1], Fergus Quilligan[2], Carlos García-Santiago[3], and Asier González-González[3]

[1] Intel Labs Europe, Intel Ireland, Leixlip, Co Kildare, Ireland
connor.upton@intel.com
[2] Irish Centre for Manufacturing Research, Collinstown Ind. Estate, Leixlip, Co Kildare, Ireland
fergus.quilligan@icmr.ie
[3] Tecnalia, Optima Unit, Parque Tecnológico de Álava, Miñano, Spain
{carlosalberto.garcia,asier.gonzalez}@tecnalia.com

Abstract. The introduction of energy efficiency as a new goal into already complex production plans is a difficult challenge. Decision support systems can help with this problem but these systems are often resisted by end users who ultimately bear the responsibility for production outputs. This paper describes the design of a decision support tool that aims to increase the interpretability of decision support outputs. The concept of 'grey box' optimisation is introduced, where aspects of the optimisation engine are communicated to, and configurable by, the end user. A multi-objective optimisation algorithm is combined with an interactive visualisation to improve system observability and increase trust.

Keywords: visualisation, optimisation, energy efficiency, manufacturing.

1 Introduction

Energy efficient manufacturing is a key research challenge for both industry and academia. Systemic energy waste is closely tied to strategic production decisions and therefore poses a complex operations-research problem. An example of this involves switching idle machine into a low-power mode. While this strategy is an effective way to save energy, it is not a straightforward task in many industrial environments. Energy savings are often subservient to production targets and decisions about changing machine states involve weighing up a complex set of goals and constraints. These include hard metrics such as production capacity, predicted inventory and product priorities as well as soft constraints such as technician skill level, engineering requests and machine recovery risks. Operations managers currently apply human expertise to cope with this complexity. In high product mix factories this problem can become very challenging and even before energy-saving is considered. Optimisation algorithms can be applied to reduce the problem space associated with this decision and to highlight energy saving opportunities; however an algorithmic approach is challenged by soft constraints and unpredictable changes in goals. In addition operations managers tend to be wary of decision support tools due to their perceived brittleness and lack of transparency [1].

C. Emmanouilidis, M. Taisch, D. Kiritsis (Eds.): APMS 2012, Part I, IFIP AICT 397, pp. 88–95, 2013.

A potential solution to this is to treat the human operators and automated systems not as autonomous agents but as team members in a joint-cognitive system [2]. Joint cognitive systems, where responsibility for control is shared between human and machine 'intelligence', are becoming increasingly important in all modern workplaces. System *observability* plays a critical role in the success of joint cognitive control as it ensures that human and automated agents can co-ordinate their actions and collaborate effectively [3]. This paper describes the design of a Decision Support System (DSS) using “grey box” optimisation and an interactive visualisation to ensure observability. This approach aims to expose aspects of an optimisation engine to increase flexibility in terms of goal and constraint settings and to communicate outputs in a manner that are easily interpretable to the end user. In this manner end user trust and the overall effectiveness of the system will be improved.

2 Applied Use Case

An individual operation within a manufacturing production process was selected for this research. This operation supports multiple products and involves a large fleet of parallel machines. Each machine requires a manual product configuration (a set-up) before processing can occur. This means that the production capacity for each product can be changed in response to demand. As well as set-ups and processing, machines may be in idle, maintenance, engineering, down or powersave states. The optimisation problem investigated here involves allocating machines to states over time under multiple constraints.

Some constraints are hard e.g. meeting production targets by specified date, while others are soft, e.g. technician skill level. An adaptation of the Cognitive Work Analysis framework [4] was used as a requirements engineering technique to understand how operations managers currently access information, prioritise goals and communicate decisions. Initially the supervisors answered questionnaires, followed by interviews and observations. Additional questionnaires were provided to further analyse the work flow and to complete task analyses. Supervisors decision making strategies were analysed using the think-aloud protocol [5] and the supervisors actions and thoughts were saved for extended analysis using audio and screen capture software. During interviews and observations three critical requirements of the solution were identified.

1. Trust

   A key challenge with any automated support system is that the final responsibility lies with the human agent. As a result an end user may not respond to a suggestion that they do not fully understand if the possible consequences are severe. Trust in automated systems is a well-known challenge in DSS and observability of system constraints is of key importance to increase user’s confidence in the system. The representation of optimisation outputs in a relevant, interpretable format will be critical. In addition, to overcome the perceived brittleness of optimisation engines it is important that the end user can view, assess and edit goals, constraints and rules that act as inputs to the system.

2. Speed
   Responsiveness is another key factor [6]. If a user is to interact and modifying goals and constraints it is important that they can get feedback on the impact of these changes without a prolonged waiting period. Furthermore, as this tool aims to optimise a current schedule it is critical that responses are provided before the current state changes.
3. Accuracy
   The nature of this dynamic scheduling problem favours a satisficing [7] approach over maximal optimisation. On top of the speed requirement, it is important to realise that goals can change, different scheduling strategies can be used to arrive at the same outputs and some constraints may not be available to the system and will require adjustments by the end user.

At a high level the main objectives are to optimise energy efficiency by maximising the time that machines can remain in power-save state, without compromising production goals. The impact of the system will be to optimise for these key performance indicators resulting in maximised output, minimised cost and increased machine utilisation.

3 Optimisation Approach

3.1 Optimisation Engine Design and User Interaction

The optimisation engine has been designed to allow for a close integration with the final user. The visualisation and user interaction requirements have heavily affected the design approach taken for the optimisation engine.

One of these main requirements is increasing the trust from the user in the results obtained from the optimisation engine. This has been addressed in two different ways, affecting several basic components of the optimiser. First, the user is able to influence the solution selection by inputting his preferences, because depending on the dynamic situation at the plant the user will prefer different alternatives. To provide these options, the optimisation engine follows 2 different algorithms. In the first one the user provides his preferences “a priori”, before the optimisation starts. This input will then be used by the algorithms as parameters that will guide the search for the optimum solution that best uses it.

The second approach does not need this prior info. In this case the optimisation algorithm will search for different solutions, together forming what is known as a “Pareto front” (the set of all solutions that are nondominated with respect to each other [8]). In this set of solutions, no one can be selected as better than another until the user selects one of them, based on his current preferences. This selection involves a certain amount of decrease in optimality in one objective to achieve a gain in another. The maximum number of solutions presented should be limited so as not to overload the supervisor and allow him to concentrate on a few possibilities. The user interview has shown that a maximum of 3 options is needed. The optimisation engine uses this requirement to select 3 solutions among the whole Pareto front that are different enough to provide a significant diversity of options to the user. The algorithm being currently under testing is the NSGA-II [9] which uses an elitist, non-domination approach, together with a crowded-comparison operator. There are plans to test other algorithms in the future, like for example Harmony Search.

The other design decision to increase trust has been using a simulation-based optimisation so the user gets detailed info on all the consequences/implications of the proposed solution. For example, machine utilisation, energy savings per machine, % of committed production in time, utilisation of tools in machines, etc. A discrete-event simulation is used, which provides a descriptive environment for such high-complexity systems as the one under study [10]. In this way the system observability and transparency is greatly increased. Figure 1 shows a high-level abstraction scheme of the DSS.

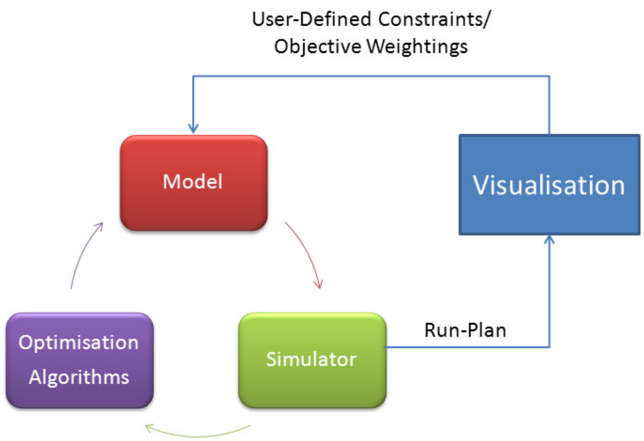

Fig. 1. Optimisation Engine - Visualisation Relation

This simulation design approach also opens the possibility to use this run-plan as a base for the introduction of new user-defined constraints directly over it. The user can introduce particular changes by using drag and drop or he can introduce them as general rules. For example, the user may want to select any machine to do some engineering work, but does not care which one. Then the optimisation engine will decide which machine to select.

The main features of the proposed optimisation engine are as follows:

- Multi-Objective. In a realistic manufacturing environment the final user has to balance different, often conflicting goals. This issue is explained below in more detail.
- Use of meta-heuristic algorithms e.g. simulated annealing and genetic algorithms. These types of techniques are approximated, meaning that they are not able to always produce the optimal solution, but have the potential to produce good solutions in a short time.
- Simulation-based: the optimisation algorithm relies on a simulation of the process to provide the fitness values for each solution. An advantage of this approach is

that it makes possible to provide the final user with increased environment descriptiveness, increasing the trust.
- Integration of PPIs (Production Performance Indicators) that act as goals and evaluation criteria for both the optimisation engine and the end user. PPIs are contextual information that provides common ground for human and automated agents.
- The user–defined constraints pose a problem of dynamic rescheduling, with the added difficulty that the time to get a solution adapted to the new scenario is limited.

To provide a solution to the new problem, the initial solution used for the rescheduling problem is obtained from the last solution generated for the old problem. This new starting solution must be first updated to the new problem, by deleting the operations already started. Starting with an already "good" solution, together with the reduction in the search space, will decrease the runtime needed to provide the new solution [11]. It is also necessary to adjust different algorithm parameters, like for example crossover operator and mutation rate in the case of genetic algorithms, to adapt to the new situation.

3.2 Multi-Objective Optimisation

This problem is defined as a multi-objective optimisation problem (MOO). There is no single global optimum solution in MOO problems. Instead of a single solution, there are a set of optimal valid solutions with different objectives magnitudes (fitness values) [12].

A general MOO problem can be defined as:

$$\text{Minimise:} \quad F(x) = [f_1(x), f_2(x) \ldots f_n(x)]$$

$$\text{Subject to:} \; G_j(x) \leq 0, j = 1, 2 \ldots m$$

where F(x) is the set of n objective functions, x is the vector of decision variables and G_j are the m independent constraints.

In this use case the two following objectives are considered:

- Maximise duration in power saving state.
- Minimise the number of machine setups.

Multiple constraints are present in the model. Among the most important we can list the following:

- Meet production target. This is the hardest constraint in the model, involving a given number of different product types at a predefined time.
- Inventory arrival time.
- Machine setups require product-specific collateral equipment, of which there is a limited amount.
- User-Defined constraints: the user is able to change constraints online through the visualisation, such as fixing the state of a machine in a specified time.

The constraints impose limits in what is called the “feasible space”, or the set of valid solutions that meet all constraints. The optimisation algorithms then try to find the Pareto front that optimises all objectives subject to those constraints.

In addition, the optimisation engine should be able to suggest maintenance timing within a defined window.

4 Visualisation Approach

The output of the optimisation algorithm is a schedule of machines transitioning between states across a production period. This schedule is presented to the user in the form of a run-plan (outline shown in Figure 1). Different machine are listed top to bottom on the left of the display. Machine states are shown over time from left to right and are colour coded so that idle states are highly salient and demand attention. The user can zoom into different time periods to assess production schedules across a shift, a day or an entire week. The current time is clearly marked and past events are shaded out to indicate that they are non-editable. On loading the screen, the user is presented with the current schedule but this may be manually edited or re-optimised by clicking on the appropriate function buttons located above and below the runplan. User defined rules may be generated based on an expert users knowledge. For example a user may wish to spread out machine setups due to an unexpected resource constraint. This is achieved by simply dragging the setup event forward or backward

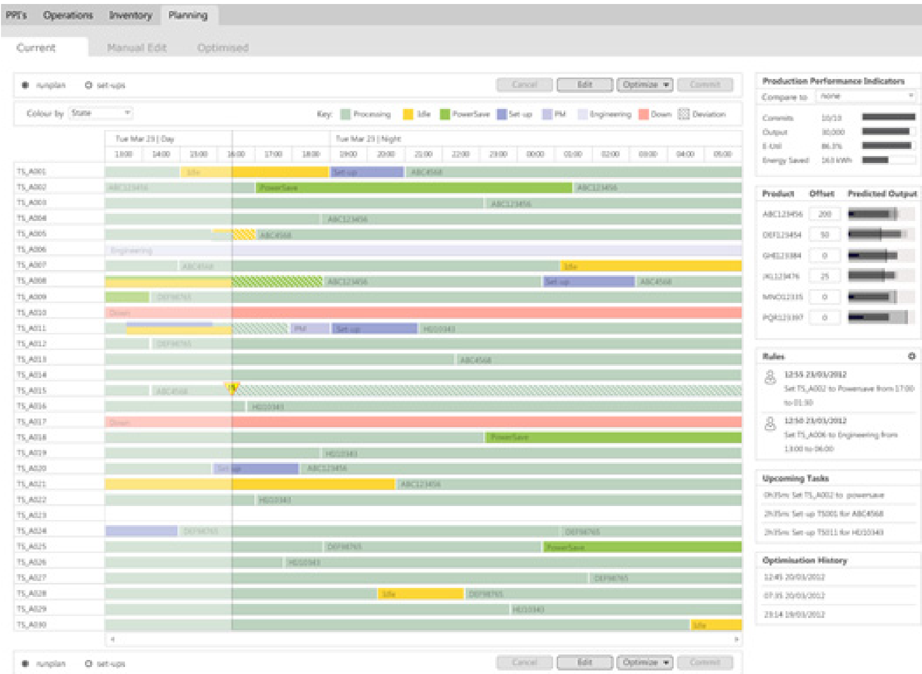

Fig. 2. Run plan Visualisation

in time. Similar actions can also be carried out to input or modify maintenance or power saving states. The panels on the right of the run-plan communicate important production performance indicators. These are updated as the user modifies the run plan in order to allow them to assess the impact of their actions. Following updates the end user can commit the run plan or may choose to re-optimise with their newly defined constraints. The tabbed menu at the top of the screen allows the user to try out multiple runplans to assess which strategy best suits their needs.

5 Conclusions

This paper describes a decision support system (DSS) for improving the energy efficiency of production plans. The research aims to improve trust in DSS, and in turn its overall effectiveness, by combining an optimisation engine with a dynamic visualisation that:

- Communicates the rules and results of an optimisation engine in an intuitive manner
- Allows end-users to generate dynamic constraints using simple drag and drop actions
- Supports experimentation using what-if scenario planning
- Generates a production plan incorporating energy saving opportunities

This concept applies principles from joint cognitive system research to support complex decision making through the combination of human flexibility with computational power. A functional prototype has been developed and future work will focus of improving the performance of the algorithms, integration with a live production system and evaluation with end users.

Acknowledgements. The presented work has been funded by the European Union FP7 "Factories of the Future" research project "KAP: Knowledge, Awareness and Prediction". The authors would like to acknowledge the support and contributions of KAP industrial and academic partners in the development of this work. We also extend our thanks to the Irish Centre for Manufacturing Research (www.icmr.ie) which is supported by Enterprise Ireland and the Industrial Development Authority of Ireland.

References

1. Carlsson, C., Turban, E.: Introduction: DSS: Directions for the next decade. Decision Support Systems 33(2), 105–110 (2002)
2. Hollnagel, E., Woods, D.D.: Joint cognitive systems: Foundations of cognitive systems engineering. Taylor and Francis, Boca Raton (2005)
3. Christoffersen, K., Woods, D.: How to make automated systems team players. Advances in Human Performance and Cognitive Engineering Research 2(1), 12 (2002)

4. Vicente, K.J.: Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work. Erlbaum and Associates, Mahwah (1999)
5. Lewis, C.: Using the 'thinking-aloud' method in cognitive interface design, IBM Research Report RC 9265, 2/17/82 IBM T. J. Watson Research Center, Yorktown Heights, NY (1982)
6. Shneiderman, B.: Computing Surveys, vol. 16(3) (1984)
7. Simon, H.A.: Rational choice and the structure of the environment. Psychological Review 63(2), 129–138 (1956)
8. Martínez-Iranzo, M., Herrero, J.M., Sanchis, J., Blasco, X., García-Nieto, S.: Applied Pareto multi-objective optimization by stochastic solvers. Engineering Applications of Artificial Intelligence 22, 455–465 (2009)
9. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
10. C.B., Beham, A., Heavey, C.: A comparative study of genetic algorithm components in simulation-based optimisation. In: Winter Simulation Conference 2008, pp. 1829–1837 (2008)
11. Bierwirth, C., Mattfeld, D.C.: Production scheduling and rescheduling with genetic algorithms. Evolutionary Computation 7, 1–17 (1999)
12. Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering. Structural and Multidisciplinary Optimization 26(6), 369–395 (2004)