

Graphical Abstract

Actor-based Large Neighborhood Search for weekly maintenance scheduling

Christian Brunbjerg Jespersen, Thomas Jacob Riis Stidsen, Kristoffer Siggaaard Wernblad, Kasper Barslund Hansen, Jingrui Ge, Simon Didriksen, Niels Henrik Mortensen

Highlights

Actor-based Large Neighborhood Search for weekly maintenance scheduling

Christian Brunbjerg Jespersen, Thomas Jacob Riis Stidsen, Kristoffer Siggaaard Wernblad, Kasper Barslund Hansen, Jingrui Ge, Simon Didriksen, Niels Henrik Mortensen

- How to allow direct and real-time integration into an optimization process?
- How to perform optimization in a real-time changing parameter space?

Actor-based Large Neighborhood Search for weekly maintenance scheduling

Christian Brunbjerg Jespersen, Thomas Jacob Riis Stidsen, Kristoffer Sigsgaard Wernblad, Kasper Barslund Hansen, Jingrui Ge, Simon Didriksen, Niels Henrik Mortensen

^a*Technical University of Denmark, Anker Egelundsvej 1, Kongens Lyngby, 2800, Hovedstaden, Denmark*

^b*Technical University of Denmark, Anker Egelundsvej 1, Kongens Lyngby, 2800, Hovedstaden, Denmark*

^c*Technical University of Denmark, Anker Egelundsvej 1, Kongens Lyngby, 2800, Hovedstaden, Denmark*

^d*Technical University of Denmark, Anker Egelundsvej 1, Kongens Lyngby, 2800, Hovedstaden, Denmark*

^e*Technical University of Denmark, Anker Egelundsvej 1, Kongens Lyngby, 2800, Hovedstaden, Denmark*

^f*Technical University of Denmark, Anker Egelundsvej 1, Kongens Lyngby, 2800, Hovedstaden, Denmark*

^g*Technical University of Denmark, Anker Egelundsvej 1, Kongens Lyngby, 2800, Hovedstaden, Denmark*

Abstract

Several problems facing the operations research field have proven difficult to solve due to their inherent uncertainty and highly dynamic nature. Stochastic optimization, fuzzy logic, and robust optimization are some of the methods that have been proposed to solve these issues. These methods make an implicit assumption on static data and a static problem setting. Maintenance scheduling is one such problem where the best available information continually updates and then therefore the scheduling continuously needs to be updated. Maintenance scheduling is a complex process and its correct implementation is usually more associated with operation management, but this paper will argue that it is possible to implement general maintenance scheduling approaches if the solution method is designed to be integrated into a business process of the kind that are usually developed by the principles of operation management.

This paper proposes a novel optimization method that is capable to opti-

mizing a scheduling problem in the following setting: primary data source is changing in real-time; external inputs affects the optimization process; multiple actors are making interdependent decision whose objectives may differ significantly. The proposed solution approach is an actor-based implementation of the large neighborhood search metaheuristic and the paper will show that this approach can naturally model the dynamic nature of operational problems.

Keywords: Large Neighborhood Search, Actor Framework, Real-time Optimization, Human-centered Computing, Interactive Systems and Tools, Decision Support Systems, Interactive Optimization.

1. Introduction

Maintenance scheduling is a dynamic and operational problem and have proven hard to solve and study in operation research due to the need of tight integration with tacit knowledge of decision makers and the way that industry usually assigns responsibility for decision-making to an individual representing only a small part of the complete process.

These multiple smaller processes are often difficult to map to a single mathematical model describing the whole system as elaborated by ((Barthélemy et al., 2002)). Solving operation research problems that are operational in nature have additional requirements over conventional static problems: they have to be responsive to changing parameters; able to be assimilated into the decision-makers workflow; allow for integration with dynamic data sources such as databases and RESTapi (Meignan et al., 2015). Operational aspects of operation research, as opposed to higher level strategic and tactical aspects, are characterized by extensive amounts negotiation and feedback on proposed schedules. The lack of integration and responsiveness can lead to schedules that are not directly implemented in practice but instead provides initial suggestions Meignan et al. (2015), which are then iterated else where in the scheduling process. In (Barthélemy et al., 2002) the authors argue that many problems that operation research aim to solve are often composed of a group of individuals whose decisions are consolidated into an "epistemic subject" for which a mathematical model can be formulated and solved, with many scheduling problems being good examples. Furthermore, some multi-objective optimization problems are a product of there being multiple actors in the decision making process each with different views on an optimal sched-

ule from their vantage point rather than there being actual multiple objectives for each individual actor.

This paper proposes a solution method that will allow for real-time optimization based on actor/user interaction and connection to a dynamic data source, effectively meaning changes to the parameter space. The proposed solution method will be tested on the multi-compartment multi-knapsack problem (MCMKP) on a large dataset from a maintenance performing company. The MCMKP naturally models what in the practical maintenance is called the weekly schedule, taken form (Palmer). It should be noted that the scientific maintenance scheduling literature deviates significantly from its practical implementation which is detailed in (Palmer). The solution method will by based on the large neighborhood search (LNS) metaheuristic. This meta heuristic was chosen due to its properties of naturally being able to work with and fix infeasible solutions and its state of the art performance on various scheduling problems.

The paper is divided into four different sections. Section 2 explains the weekly maintenance scheduling model in detail and forms the foundation of the paper. Section 3 shows that results coming from the implemented system where the implementation will be affected by simulated user-interaction. Section 4 will discuss the implications of the research and possible future research directions.

1.1. The Weekly Schedule: Multi-compartment Multi-knapsack Problem with capacity penalties

The actor-based large neighborhood search is implemented on the MCMKP which models that weekly schedule in maintenance. The model is comprised of five different sets. P is the number of weekly periods; W is the number of work orders; τ is the number of different traits; E is a set that defines which work orders should be excluded from a specific weekly period; I is an inclusion set that defines the allocation of specific work orders which should be included in a specific weekly period. The model has four parameters. v_{pw} is the value of work order w in weekly period p ; d is the penalty for exceeding a specific trait capacity; $c_{w\tau}$ is the capacity requirement for work order w for trait t ; $cap_{p\tau}$ is the total amount of capacity available in for weekly period p for for trait t . The model has 2 decision variables. x_{wp} is a binary decision variable equal to one if work order w is in weekly period p and zero otherwise; $pen_{p\tau}$ is non-negative decision variable equal to the amount of excess capacity above the $cap_{p\tau}$ in weekly period p for trait τ . The parameters v ,

cap , Q , and P are functions of time, τ , in this case as they will be subject to change during the solution process.

$$\text{Min} \quad \sum_{w=1}^W \sum_{p=1}^P v_{wp}(t) \cdot x_{wp}(t) + \sum_{p=1}^P \sum_{\tau=1}^T d \cdot pen_{p\tau}(t) \quad (1)$$

subject to:

$$\sum_{w=1}^W c_{w\tau} \cdot x_{wp}(t) \leq cap_{p\tau}(t) + pen_{p\tau}(t) \quad \forall p \in P, \forall \tau \in T \quad (2)$$

$$\sum_{w=1}^W x_{wp}(t) = 1 \quad \forall p \in P \quad (3)$$

$$x_{wp}(t) = 0 \quad \forall (w, p) \in E(t) \quad (4)$$

$$x_{wp}(t) = 1 \quad \forall (w, p) \in I(t) \quad (5)$$

$$x_{wp}(t) \in \{0, 1\} \quad \forall w \in W, \forall p \in P \quad (6)$$

$$pen_{p\tau}(t) \in \mathbb{R}^+ \quad \forall p \in P, \forall \tau \in T \quad (7)$$

The objective function 1 minimizes the total weight of all work order assignments together with the penalty d for exceeding the capacity given in constraint 2. Constraint 2 ensures that all the weights $c_{w\tau}$ for each activity in an work order, given that it has been assigned, is lower than the capacity for each period and for each trait τ . $pen_{p\tau}$ is the amount of exceeded capacity that is needed for the current assignment of work order to be feasible. Constraint 3 makes sure that each work order is assigned to atleast a single period. Constraint 4 excludes a work order from a certain period and constraint 5 forces a specific work order to be in a specific period. Constraint 6 and 7 specify the variable domain for x_{wp} and $pen_{p\tau}$ respectively. The effects of changing E , I , cap , and v in real-time will be examined to determine their effects on the weekly schedules and objective value.

2. Solution Method

2.1. Actor-based Large Neighborhood Search

A problem which is affected by user-interaction and requires real-time feedback needs an optimization approach that is able to repair infeasible schedules and while also converging quickly. For this the large neighborhood search metaheuristic has been shown satisfy these requirements in the literature Gendreau and Potvin (2019).

The LNS metaheuristic is defined for static problems, meaning that the parameters that make up the problem instance is not subject to change after the algorithm has been started. To make the LNS able adapt to changing parameters in real-time a message system have been implemented into the existing framework. This extension is shown in algorithm 1.

2.1.1. Messages And Destructors

LNS in its most basic form has one constructor and one destructor which repeatedly destroy and rebuild the schedule. For the AbLNS we will generalize on this concept by including messages as destructors of the classic LNS implementation. This generalization can be seen as being somewhat similar to how the adaptive LNS (ALNS) is formulated, but where the different constructors and destructors are chosen externally as well.

Extending on the classic setup we define the following set of destructors, M :

- m_1 : Inclusion destruct message
- m_2 : Exclusion destruct message
- m_3 : Capacities destruct message
- m_4 : Weights destruct message
- m_5 : Random destruct message

Each of these messages affect different parts of the MCMK problem (weekly schedule). Notice here that the first four messages destruct the solution by changing the parameter space and the last message is a random destructor.

Generalizing the destructors from being static structures into messages allows the solution to change in real-time to a changing parameter space

meaning that the algorithm does not need to restart to handle changes in data.

Algorithm 1 Actor-based Large Neighborhood Search

```

1: Input queue = message queue
2: Input P = problem instance
3: Input x = initial schedule
4: while true do
5:   while queue.has_message() do
6:      $P.update(m)$ 
7:      $x.destruct(m)$ 
8:   end while
9:    $x^t = x.repair()$ 
10:  if  $c(x^t) < c(x)$  then
11:     $x = x^t$ 
12:    queue.send(x)
13:  end if
14:  queue.push( $m_5$ )
15: end while
16: return  $x^b$ 

```

The basic LNS setup have here been extended with a ‘message queue’. This message queue will be read from on every iteration of the LNSs main iteration loop. Here we notice that the incoming message are able to change both the solutoin but also the problem instance itself. Here we see one of the defining features of the LNS metaheuristic in play, that due to its inherrent property of being able to optimize a solution that have become infeasible which is something that is very likely to happen when you change the parameter of the problem instance itself.

Another less obvious property the message queue allows is for the algorithm to run indefinitely and instead of restarting the algorithm you instead pass messages to it to allow it be adjust both the solution space and the parameter space. This property avoid the issue of time consuming initial convergence as the algorithm will be found in an optimal state when the solution is perturbed.

3. Results

The results section will: 1. introduce the real-world data instance; 2. show the effect of forcing item set in the specific weekly schedules; 3. show the effect of changing the period capacities, and 4. show the effect of dynamically changing the value of the work orders v .

3.1. Data Instance

	Number of Item Sets	Number of Compartments	Number of Knapsacks
Instance 1	3487	16	52

Table 1: Table Caption

3.2. Response to Inclusion

The response to the inclusion of a work order is given by I parameter of the model which is constrained in 5 of model given in 1.1.

The inclusion is made of forcing certain allocations of work orders to be in specific periods. Below a table is provided to show what changes will occur and at what and at what point in time.

	At Time: 01:00	At Time: 02:00	At Time: 03:00	At Time: 04:00	At Time: 05:00
$\Delta P $	10	20	30	40	50

With the inputs defined we will explain the main results which are shown in the figure below.

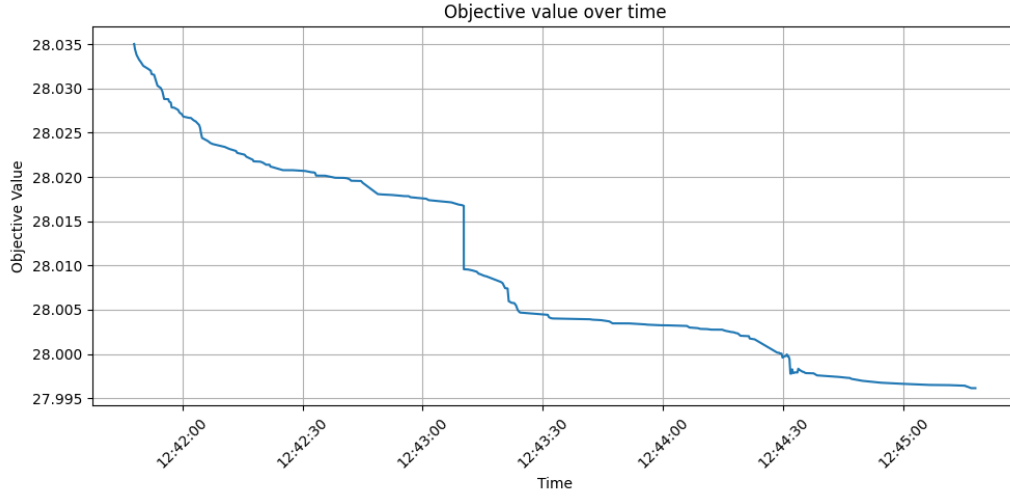


Figure 1: Figure Caption

3.3. Response to Exclusion

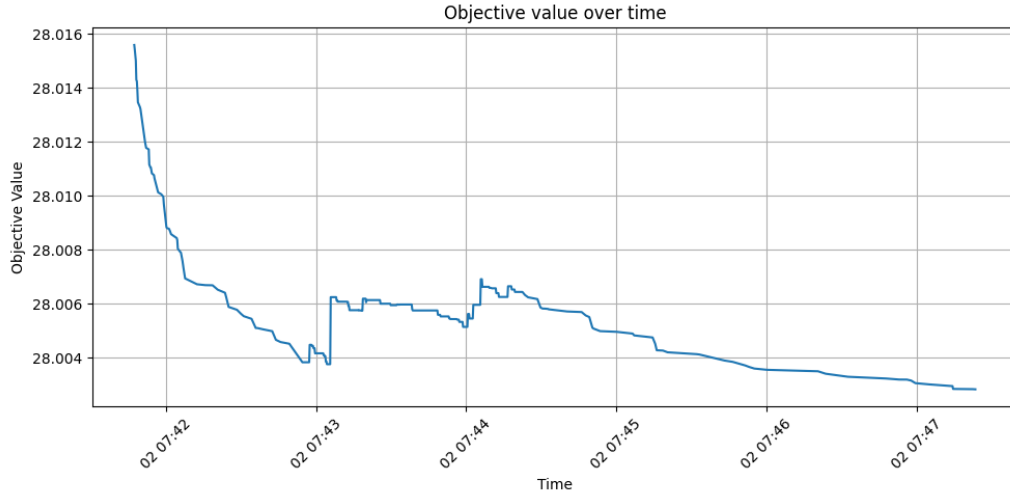


Figure 2: Figure Caption

3.4. Response to Changes in Knapsack Capacities

The effects of changes to capacities will be illustrated in the same way as it was with the response to inclusion and below we see the table that shows which inputs that the AbLNS will be affected by.

	At Time: 01:00	At Time: 02:00	At Time: 03:00	At Time: 04:00	At Time: 05:00
$\Delta p $	16	16	16	16	16
$\Delta \tau $	16	16	16	16	16
$\Delta cap $	100	200	400	800	1600

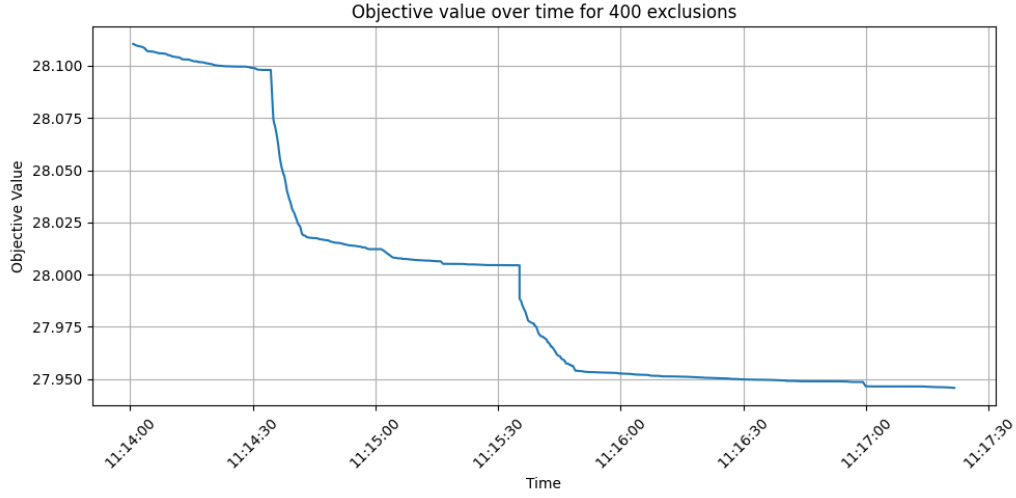


Figure 3: Figure Caption

Correspondingly we also have the figure below in which the resources are decreasing.

3.5. Response to Changes in Item Weights

The final parameter that will be changed is the work order value v . This section will be more elaborate as we have to show how that the work orders are rearranged due to the changes in their value across the different periods.

	At Time: 01:00	At Time: 02:00	At Time: 03:00	At Time: 04:00	At Time: 05:00
$\Delta w $	20	40	80	160	320
$\Delta p $	26	26	26	26	26
$\Delta v $	$1 \cdot 10^5$	$2 \cdot 10^5$	$4 \cdot 10^5$	$8 \cdot 10^5$	$1.6 \cdot 10^6$

4. Discussion

4.1. Integration

Figure 4 illustrates a classic approach to operations research where a model allows the decision maker to obtain a solution to from which to make better informed decision. This approach is suitable to static problems and problems where the parameters that make up the problem change slowly. This approach is often unsatisfactory in environments where the best available information is ever changing.

Actor-based large neighborhood search enables tight integration between both the users and a dynamic data sources but crucially it naturally allows models to communicate with each other mimicking most practical decisions which are made up of multiple actors, as in the maintenance scheduling problem.

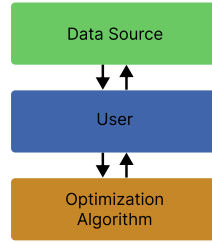


Figure 4: Traditional use case of developed optimization algorithms. Notice that the optimization algorithm is separated from the data source, requiring user interaction to optimize the process

Figure 5 illustrates the setup that is enabled by using an actor-based optimization approach. Due to the implemented message system it becomes possible to build large decision making structures from smaller units.

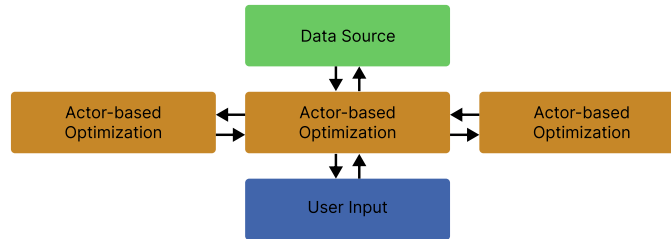


Figure 5: Figure Caption

The three most consequential properties of this approach is: 1. continual optimization saves the initial time it takes to reach convergence; 2. due to the message system the optimization approach will always be responsive to changes in both the parameter space and the solution space; 3. due to the encapsulation and message passing properties of the suggested optimization approach it becomes possible to apply the metaheuristic in multi-model and hierarchical model setups, providing an approach for modelling and optimizing large scale systems.

4.2. Continuous Optimization

By continually optimizing it becomes possible to only optimize the perturbations to the schedule. This means that we avoid having to optimize the problem to reach initial convergence.

In many cases this can save a significant amount of computations, refer to alza et al. (2023), especially if specialized constructors and destructors are implemented to handle the specific perturbations.

4.3. Message Passing versus Restarts

There is evidence that the effect of dynamic optimization is not always efficient, alza et al. (2023) mentions the idea that dynamic optimization approaches can be "elusive" in that they do not always provide a speedup over restarting the solution method implementation. The paper clearly shows that dynamic approaches are the most effective when changes are small and frequent, which does align with the idea behind the actor-based LNS in that changes should be optimized around when ever they occur. T

4.4. System Level Optimization

This approach could also enable larger scale optimizations providing a modular approach to operations research. This is beneficial

«««< HEAD

References

alza, j., bartlett, m., ceberio, j., mccall, j., 2023. on the elusivity of dynamic optimisation problems. swarm and evolutionary computation 78, 101289. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2210650223000627>, doi:10.1016/j.swevo.2023.101289.

- Barthélemy, J., Bisdorff, R., Coppin, G., 2002. Human centered processes and decision support systems. *European Journal of Operational Research* 136, 233–252. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0377221701001126>, doi:10.1016/S0377-2217(01)00112-6.
- Gendreau, M., Potvin, J.Y. (Eds.), 2019. Handbook of Metaheuristics. volume 272 of *International Series in Operations Research & Management Science*. Springer International Publishing, Cham. URL: <http://link.springer.com/10.1007/978-3-319-91086-4>, doi:10.1007/978-3-319-91086-4.
- Meignan, D., Knust, S., Frayret, J.M., Pesant, G., Gaud, N., 2015. A Review and Taxonomy of Interactive Optimization Methods in Operations Research. *ACM Transactions on Interactive Intelligent Systems* 5, 1–43. URL: <https://dl.acm.org/doi/10.1145/2808234>, doi:10.1145/2808234.
- Palmer, R.D., . Maintenance Planning and Scheduling Handbook, 4th Edition. 4th edition ed., McGraw Hill.

=====

References

- alza, j., bartlett, m., ceberio, j., mccall, j., 2023. on the elusivity of dynamic optimisation problems. *swarm and evolutionary computation* 78, 101289. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2210650223000627>, doi:10.1016/j.swevo.2023.101289.
- Barthélemy, J., Bisdorff, R., Coppin, G., 2002. Human centered processes and decision support systems. *European Journal of Operational Research* 136, 233–252. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0377221701001126>, doi:10.1016/S0377-2217(01)00112-6.
- Gendreau, M., Potvin, J.Y. (Eds.), 2019. Handbook of Metaheuristics. volume 272 of *International Series in Operations Research & Management Science*. Springer International Publishing, Cham. URL: <http://link.springer.com/10.1007/978-3-319-91086-4>, doi:10.1007/978-3-319-91086-4.

Meignan, D., Knust, S., Frayret, J.M., Pesant, G., Gaud, N., 2015. A Review and Taxonomy of Interactive Optimization Methods in Operations Research. *ACM Transactions on Interactive Intelligent Systems* 5, 1–43. URL: <https://dl.acm.org/doi/10.1145/2808234>, doi:10.1145/2808234.

Palmer, R.D., . Maintenance Planning and Scheduling Handbook, 4th Edition. 4th edition ed., McGraw Hill.

»»»> e7eda4191ef254a36c2a136d7e87517183270cac