Integrating a temporal component into multi-criteria majority-rule sorting models

Application to cyber defence decision aiding

Arthur Valko[1], **Alexandru-Liviu Olteanu**[2], **David Brosset**[3] and **Patrick Meyer**[4]

Abstract. In certain applications, evaluations of decision alternatives on multiple criteria may vary in time. This is the case in the cyber-defence setting, where multi-criteria decision aiding can be used to help a decision maker to evaluate multiple actions on a system under attack, in order to return it to an operational state. More specifically, these actions might have different effects on the future availability and efficiency of various functionalities of the system. This is why, in this work, we tackle the problem of integrating a temporal component into multi-criteria majority-rule sorting models. We choose to integrate this component by building a hierarchical sorting model, which has the advantage of being easily understandable by the decision maker, while helping him to elicit the model parameters.
Keywords: multi-criteria decision aiding, sorting, outranking relation, mixed-integer programming, times series

1 Introduction

Multi-criteria Decision Aiding (MCDA) is the study of decision problems, methods and tools which may be used in order to assist a decision maker in reaching a decision when faced with a set of decision alternatives, described via multiple – often conflicting – properties or criteria [8]. Usually, three types of decision problems are put forward in this context: the *choice problem*, which aims to recommend a subset of alternatives, as restricted as possible, containing the "satisfactory" ones; the *sorting problem*, whose goal is to assign each alternative into predefined ordered categories; the *ranking problem*, which orders the alternatives by decreasing degree of preferences. Various models have been proposed to support decision makers facing a multi-criteria decision problem [2] and to represent their preferences. Roughly speaking, they originate from two methodological schools: first, in the *outranking* methodologies, any two alternatives are compared pair wisely on the basis of their evaluations on the set of criteria; second, methods based on *multiattribute value theory* aim to construct a numerical representation of the decision maker's preference on the set of alternatives [1].

When the evaluations of the alternatives on the various criteria vary in time (for example, a criterion takes different values for the same alternative in time), two elementary options exist to integrate this temporal component into MCDA models: first, one could duplicate the criterion which varies in time, and consider one version of the criterion for each considered time step; second, one could try to aggregate this time series on the considered criterion into a synthesis evaluation [9](for example by giving more or less importance to various time steps). The first option might lead to a huge number of criteria that the decision maker must consider, when determining the preferential parameters of the MCDA model at hand. With the second option, one might miss a large amount of information which was present in the original time series. In this article we explore a third option, which is to decompose an existing decision model into a hierarchical one, to account for the various time steps.

2 Multicriteria Decision Aiding in Cyberdefence

This work is mainly motivated by a practical context linked to cyber-defence. Following a cyber-attack, a system may be in a vulnerable state, and many of its elementary functions may be in a degraded state or completely unavailable [3]. To our knowledge, multi-criteria decision aiding has only been poorly applied to the cyber-defence context (see, e.g. [4]). In our naval applicative context, where the system at hand is a ship, our goal is to help a decision maker evaluate the various actions which will allow, following a cyber-attack, to restore some or all of the ship's functions.

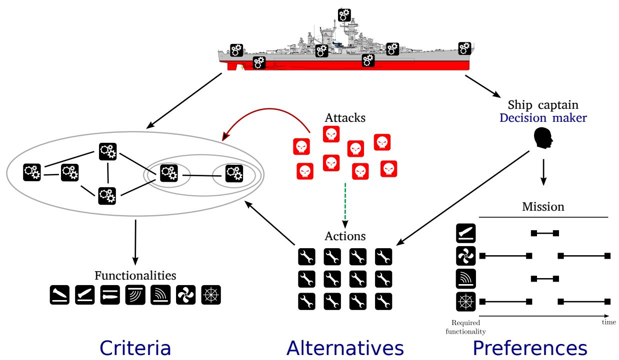

Figure 1. Global ship defense process

Figure 1 illustrates our use-case. A ship is composed of several information systems and PLCs (Programmable logic controllers), which may be divided into different sub-systems and connected to each other and to the outside world. These systems contribute to attaining the various functionalities of the ship (as, e.g., detection using radars, movement using the propulsion and steering systems, defense using the weapons systems, ...). The captain uses the ship's different functionalities in order to carry out the mission at hand. Depending on the progress of the mission, the importance of some of the ship's functionalities will vary across time [5].

As with any system, computer attacks can be carried out on the ship's various subsystems. These attacks then compromise the proper functioning of the ship and the progress of its mission [6]. To defend

[1] Chair of Cyber defence of naval systems, email: arthur.valko@ecole-navale.fr
[2] Lab-STICC, Université Bretagne Sud, Lorient, France, email: alexandru.olteanu@univ-ubs.fr
[3] Naval Academy Research Institute, Chair of Cyber defence of naval systems, email: david.brosset@ecole-navale.fr
[4] IMT Atlantique, Lab-STICC, Univ. Bretagne Loire, F-29238 Brest, France, email: patrick.meyer@imt-atlantique.fr

the ship, several actions are possible in order to stop or contain the attack. These defensive actions depend on the targeted systems and the type of attack. However, like the attacks, these actions have an impact, sometimes not negligible, on multiple systems and therefore on several of the ship's functionalities. At the same time, they may be more or less effective and must follow an intervention protocol that, in addition to taking time, is sometimes applied in stages and on several connected systems.

Consequently, we wish to assist the decision maker (in this case, the ship's captain) in the evaluation of alternatives (defensive actions), whose evaluations according to multiple criteria (ship's functionalities) change over time (mission).

This evaluation is typically ordinal (good, medium, bad), which leads us to choose a sorting method for our problem. As the effects of such an action might have various consequences in time, across multiple criteria, we are studying here the integration of the temporal component into a sorting method, called MR-Sort (for majority rule sorting) [7].

In the end, the objective of this evaluation is to propose to the decision maker a dashboard, similar to Figure 2, facilitating the choice of a proper defence action. The actions assigned to the best category would be highlighted in the dashboard (like actions 3 and 4 in the figure), while showing at the same time the state of the various functionalities of the ship as a function of time, as well as the constraints related to the mission.

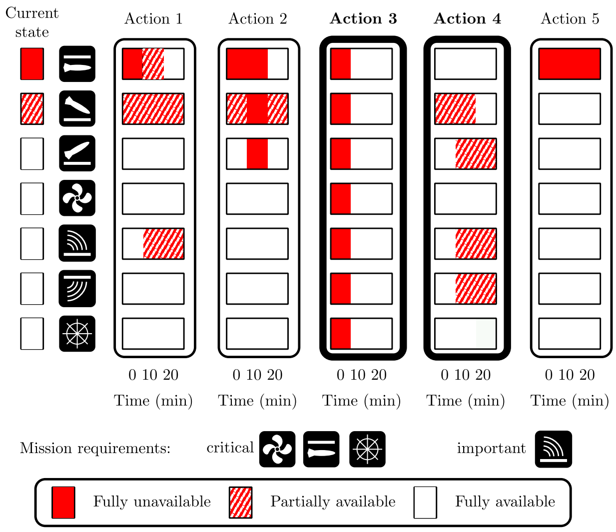

Figure 2. Decision support dashboard

For example, action 3 is a complete reboot of all of the sub-systems of the ship, which takes about ten minutes and returns the ship to a fully operational state. However, some mission-critical features will be interrupted during the first 10 minutes (complete loss of functionality depicted in red). Action 4 guarantees the operation of critical features, but limits the performance of an important feature (radar) during a long period (partial loss of functionality depicted as crosshatched). It is now up to the ship captain to select among these two actions which one is preferable in the context of the ship and the mission.

3 Integrating time into a hierachical MR-Sort model

We propose to integrate the two considered dimensions of an alternative, its criteria and their evolution in time, by using a hierarchy of MR-Sort models. First the problem is divided into sub-problems using one dimension (time-wise or criteria-wise). Each sub-problem is solved independently of the others while their results are then taken as input within a global model that handles the second dimension. While such an approach may reduce the expressivity with respect to a model taking into account both dimensions at once (a topic for a future work), the added benefit of such an approach lies in the reduced cognitive effort required by the DM as well as in its interpretability. The DM may intervene during an elicitation process or during a day-to-day use of such a model both at the global level, and also locally, considering either only one state on all criteria for a given time frame, or the evolution of a single criterion across time.

Let us consider a finite set of alternatives A, a finite set of time step indexes $T = \{1, ..., n\}$ and a finite set of criteria indexes $J = \{1, ..., m\}$. Each alternative $a \in A$ is evaluated on any criterion and any time step through a function $g_{t,j} : A \to \mathbb{R}$, where $g_{t,j}(a)$ ($t \in T, j \in J$) denotes the performance of the alternative a at time step t and on criterion j. The alternatives are to be sorted into k categories ($c_1, \ldots, c_k$) ordered by their desirability from c_1 being the least preferred, to c_k being the ideal one. We denote the vector containing the indexes of these categories with $K = \{1, ..., k\}$.

We propose to model this sorting problem by first disaggregating the alternative evaluations using each time step, constructing MR-Sort sub-models for each of them and then aggregating the results of these sub-models into the evaluations that will be used to construct a global model leading to assignments to categories $c_1, \ldots, c_k$.

In order to keep the model general, we consider that each MR-Sort sub-model may assign an alternative to a different set of categories, which may be defined beforehand by the DM, i.e. $c_{t,1}$, $\ldots$, $c_{t,k_t}, \forall t \in T$, ordered with respect to their desirability. We denote the vectors containing the indexes of these categories with $K_t = \{1, ..., k_t\}, \forall t \in T$.

We define each MR-Sort sub-model (one for each $t \in T$) using the following parameters:

- majority threshold $\lambda_t \in]0.5, 1]$;
- criteria weights $w_{t,j} \in [0, 1], \forall j \in J$ with $\sum_{j \in J} w_{t,j} = 1$;
- category profiles evaluations $b_{t,h,j}, \forall h \in 1..k_t + 1, \forall j \in J$.

Each category $c_{t,h}, h \in K_t, t \in T$ of the t^{th} sub-model is delimited in the criteria space by a lower frontier $b_{t,h-1}$ and an upper one $b_{t,h}$. Furthermore the frontier performances are non-decreasing. i.e . $b_{t,h,j} \geqslant b_{t,h-1,j} \forall t, h, j \in T \times \{1..k_t + 1\} \times J$.

Two rules to assign an alternative to a class may be found in literature: the pessimistic and the optimistic rules. We will use the first as it is the most commonly used. An alternative a is therefore assigned to the highest possible category a outranks its lower frontier but not its upper frontier.

An alternative a is said to outrank another one (in our case a category limit) if and only if there is a sufficient coalition of criteria supporting the assertion a is at least as good as the frontier.

For each sub-model with $t \in T$, we define local concordance indices between an alternative a and a category profile $b_{t,h}$ as:

$$C_{t,j}(a, b_{t,h}) = \begin{cases} 1, \text{if } g_{t,j}(a) \geqslant b_{t,h,j}, \\ 0, \text{otherwise.} \end{cases} \quad (1)$$

The support of the coalition of criteria in favour of the outranking, also called the global concordance, which we denote with $C_t(a, b_{t,h})$ is then defined as:

$$C_t(a, b_{t,h}) = \sum_{j \in J} w_{t,j} C_{t,j}(a, b_{t,h}) \quad (2)$$

Using these previously defined global concordances, in order to determine whether they are sufficient in order to validate an outranking situation, we simply compare them to the majority thresholds λ_t. We define for each submodel the S_t relations as:

$$a \, \mathrm{S_t} \, b_{t,h} \iff C_t(a, b_{t,h}) \geqslant \lambda_t, \forall t \in T, \forall h \in 1..k_t + 1 \quad (3)$$

Using these outranking relations, we can now define the assignment rule of the MR-Sort sub-models. An alternative a is assigned to category $c_{t,h}, \forall t \in T$, when:

$$a \in c_{t,h} \iff a \, \mathrm{S_t} \, b_{t,h} \text{ and } a \, \not\mathrm{S}_t \, b_{t,h+1} \quad (4)$$

The results of these assignments, for all sub-models $t \in T$, will then form the aggregated evaluations of the alternatives, which will then be used in the top model of the hierarchy:

$$\overline{g}_t(a) = h \iff a \in c_{t,h}, \forall t \in T \quad (5)$$

We define the top MR-Sort model using the following parameters:

- majority threshold $\overline{\lambda} \in]0.5, 1]$;
- criteria weights $\overline{w}_t \in [0, 1], \forall t \in T$ with $\sum\limits_{t \in T} \overline{w}_t = 1$;
- category profiles evaluations $\bar{b}_{h,t}, \forall h \in 1..k + 1, \forall t \in T$.

Similarly to the MR-Sort sub-models, we define a local concordance index $\overline{C}_t$, a global concordance index $\overline{C}$ and an outranking relation $\overline{\mathrm{S}}$, replacing the model parameters with the ones defined above and the alternative evaluations with the ones in Equation (5). An alternative will then be assigned to a category c_h if and only if $a \, \overline{\mathrm{S}} \, \bar{b}_h$ and $a \, \overline{\not\mathrm{S}} \, \bar{b}_{h+1}$.

4 Parameters inference of a hierarchical MR-Sort model

Inferring the parameters of majority-rule sorting models from assignment examples provided by the DM usually uses mathematical programming techniques involving binary variables, such as in Leroy et al. [7]. These approaches find an optimal solution, but they may also require large amounts of computational resources and time.

Here, we follow the same path, and in order to infer the parameters of a hierarchical MR-Sort model only from the input of the DM on the final class assignment of a set of alternatives, we construct a mixed-integer linear program, which we present in Appendix A.

The first three lines contain the parameters of the model. We find the set of alternatives A', the set of criteria J, the set of time steps T, the number of categories k and a small constant γ, which is used to model strict inequalities. The alternatives evaluations and the category assignments are then given on lines 2 and 3 respectively.

Lines 4 to 14 contain the variables of the model. On the first five lines we have the variables corresponding to the parameters of the top MR-Sort model (denoted with a top bar) and those corresponding to the bottom MR-Sort models (underlined), namely the majority thresholds ($\bar{\lambda}$ and $\underline{\lambda}_t$), the criteria weights ($\bar{w}_j$ and $\underline{w}_{t,j}$) and the category profiles ($\bar{b}^h_j$ and $\underline{b}^h_{t,j}$).

On line 10 we have two variables that correspond to the local concordance indexes $\bar{C}$ between an alternative a and the lower and upper profiles of its assigned category $K(a)$. We denote $\bar{C}_t(a, b_{K(a)})$ with $\bar{C}^+_t(a)$ and $\bar{C}_t(a, b_{K(a)-1})$ with $\bar{C}^-_t(a)$. The $\bar{W}$ variables on line 11 are used in order to calculate the weighted sum, i.e. the global concordance indexes for the top model.

The $\underline{C}$ variables on line 12 represent the local concordance between an alternative and any category profile for the sub-models of the hierarchy, while the $\underline{W}$ variables on the following line are used in order to calculate the weighted sum, i.e. the global concordance indexes for each sub-model.

Finally, the x binary variables are needed in order to encode which category profile of a sub-model alternative a outranks. These variables will then be used in order to give the intermediary evaluations of a across each time step $t \in T$.

Lines 15 to 41 contain the constraints of the MIP model. The first two (15, 16) fix the sum of the weights of each model to 1. The following six constraints are needed in order to bound the evaluations of the category profiles to a $[0, 1]$ interval and also ensure the dominance constraints between them.

Constraints 23 and 24 are used to fix $C_{t,j}(a, \underline{b}^h)$ to 1 if the evaluation of a on criterion j is greater or equal to that of profile $\underline{b}^h$), and to 0 otherwise. Constraints 25 to 27 are then used to fix $W_{t,j}(a, \underline{b}^h)$ to w_j when $C_{t,j}(a, \underline{b}^h)$ is 1, and to 0 otherwise. Next we have constraints 28 and 29 which fix the x variables to 1 if a outranks profile $\underline{b}^h$ and to 0 otherwise.

With this, all of the constraints needed to model the sub-models in our hierarchy have been added. We continue with the constraints that model the top model in our hierarchy.

The constraints on lines 30 and 31, together with those on lines 32 and 33, resemble constraints 23 and 24, except that the evaluation of alternative a for a time step t is now given by the category to which a is assigned by the corresponding MR-Sort sub-model. This evaluation is simply given by the sum of the x variables. We also only express these constraints only for the top and bottom profiles of the assigned category $K(a)$ (notice the $\bar{C}^+_j(a)$ binary variables for the top profile and the $\bar{C}^-_j(a)$ binary variables for the bottom one). Since the evaluations of a for a given time step $t \in T$ are on a scale from 0 to k, we have also used a scaling factor equal to k.

The following 6 constraints are used to compute the $\bar{W}^+_t(a)$ and $\bar{W}^-_t(a)$ variables equal to w_t if a is at least as good as the corresponding category profile on time step t, and 0 otherwise.

The final two constraints impose that a does not outrank the upper profile, and that it outranks the lower profile, of its assigned category.

5 Experimental validation and preliminary results

The proposed hierarchical MR-Sort model may be used to solve the same type of problem as the original MR-Sort model. Nevertheless, as it breaks the decision problem into sub-problems, several research questions may be raised, also in relation to the mathematical programs needed to infer each type of model.

The first question concerns the complexity of the mathematical programs. In particular, the computation times required to learn each model may become a factor in applying them in practice as the problem size increases. The second question concerns the expressivity of the hierarchical model when compared to the classical MR-Sort model. Is either of the two models more expressive than the other? How many alternatives are needed in order to accurately infer each of them?

To answer these questions, a test platform was developed. The platform generates alternatives and random models of each type. Considering the models as representations of fictitious DMs, we then sample these sets of alternatives, assign them using the aforementioned models and use mathematical programs to infer new models.

With respect to the first research questions, initial results tend to show that the program for inferring the hierarchical MR-Sort model takes a significantly longer time to finish. This is due to the fact that it is more complex due to it having more binary variables.

The second question may be answered by trying to infer a hierarchical MR-Sort model using the assignment examples from a classical MR-Sort model, or vice-versa.

REFERENCES

[1] Raymond Bisdorff, Luis C Dias, Patrick Meyer, Vincent Mousseau, and Marc Pirlot, *Evaluation and Decision Models with Multiple Criteria*, Springer, 2015.

[2] D. Bouyssou, T. Marchant, M. Pirlot, A. Tsoukiàs, and P. Vincke, *Evaluation and decision models with multiple criteria: Stepping stones for the analyst*, International Series in Operations Research and Management Science, Volume 86, Boston, 1st edn., 2006.

[3] Ulrik Franke and Joel Brynielsson, 'Cyber situational awareness–a systematic review of the literature', *Computers & Security*, **46**, 18–31, (2014).

[4] Alexander A. Ganin, Phuoc Quach, Mahesh Panwar, Zachary A. Collier, Jeffrey M. Keisler, Dayton Marchese, and Igor Linkov, 'Multicriteria decision framework for cybersecurity risk assessment and management', *Risk Analysis*, (Published online 05 September 2017).

[5] Michael R Grimaila, Larry W Fortson, and Janet L Sutton, 'Design considerations for a cyber incident mission impact assessment (cimia) process', Technical report, Air Force Inst. Of Tech. Wright-Patterson AFP on Center for Cyberspace Research, (2009).

[6] Gabriel Jakobson, 'Mission cyber security situation assessment using impact dependency graphs', in *2011 Proceedings of the 14th International Conference on Information Fusion (FUSION)*, pp. 1–8. IEEE, (2011).

[7] A. Leroy, V. Mousseau, and M. Pirlot, 'Learning the parameters of a multiple criteria sorting method', in *Algorithmic Decision Theory*, eds., R. Brafman, F. Roberts, and A. Tsoukiàs, volume 6992, pp. 219–233. Springer, (2011).

[8] B. Roy, *Multicriteria Methodology for Decision Aiding*, Kluwer Academic, Dordrecht, 1996.

[9] Ming Zhou and Ye Chen, 'An aggregation framework for time series-based mcda', in *Grey Systems and Intelligent Services (GSIS), 2011 IEEE International Conference on*, pp. 571–575. IEEE, (2011).

A MILP for inferring a hierarchical MR-Sort model

Table 1: Mixed integer linear program for Hierarchical MR-Sort;

Parameters:		
A', J, T, k, γ		(1)
$g_{t,j}(a) \in [0,1]$	$\forall a \in A', \forall t \in T, \forall j \in J$	(2)
$K(a) \in \{0,...,k\}$	$\forall a \in A'$	(3)
Variables:		
$\bar{\lambda} \in [0.5,1]$		(4)
$\underline{\lambda}_t \in [0.5,1]$	$\forall t \in T$	(5)
$\bar{w}_t \in [0,1]$	$\forall t \in T$	(6)
$\underline{w}_{t,j} \in [0,1]$	$\forall t \in T, \forall j \in J$	(7)
$\bar{b}_t^h \in \{1,...,k\}$	$\forall t \in T, \forall h \in \{1,...,k\}$	(8)
$\underline{b}_{t,j}^h \in [0,1]$	$\forall t \in T, \forall j \in J, \forall h \in \{1,...,k\}$	(9)
$\bar{C}_t^+(a), \bar{C}_t^-(a) \in \{0,1\}$	$\forall a \in A', \forall t \in T$	(10)
$\bar{W}_t^+(a), \bar{W}_t^-(a) \in [0,1]$	$\forall a \in A', \forall t \in T$	(11)
$\underline{C}_{t,j}(a, \underline{b}^h) \in \{0,1\}$	$\forall a \in A', \forall t \in T, \forall j \in J, \forall h \in \{0,...,k\}$	(12)
$\underline{W}_{t,j}(a, \underline{b}^h) \in [0,1]$	$\forall a \in A', \forall t \in T, \forall j \in J, \forall h \in \{0,...,k\}$	(13)
$x(a, \bar{b}_t^h) \in \{0,1\}$	$\forall a \in A', \forall t \in T, \forall h \in \{0,...,k\}$	(14)
Constraints:		
$s.t. \sum_{i \in T} \bar{w}_t = 1$		(15)
$\sum_{j \in J} \underline{w}_{t,j} = 1$	$\forall j \in J$	(16)
$\bar{b}_t^0 = 0$	$\forall t \in T$	(17)
$\underline{b}_{t,j}^0 = 0$	$\forall t \in T, \forall j \in J$	(18)
$\bar{b}_t^k = 1$	$\forall t \in T$	(19)
$\underline{b}_{t,j}^k = 1$	$\forall t \in T, \forall j \in J$	(20)
$\bar{b}_t^{h-1} \leqslant \bar{b}_t^h$	$\forall t \in T, \forall h \in \{1,...,k\}$	(21)
$\underline{b}_{t,j}^{h-1} \leqslant \underline{b}_{t,j}^h$	$\forall t \in T, \forall j \in J, \forall h \in \{1,...,k\}$	(22)
$g_{t,j}(a) - \underline{b}_{t,j}^h + \gamma \leqslant \underline{C}_{t,j}(a, \underline{b}^h)$	$\forall a \in A', \forall t \in T, \forall j \in J, \forall h \in \{0,...,k\}$	(23)
$(\underline{C}_{t,j}^h - 1) \leqslant g_{t,j}(a) - \underline{b}_{t,j}^h$	$\forall a \in A', \forall t \in T, \forall j \in J, \forall h \in \{0,...,k\}$	(24)
$\underline{W}_{t,j}(a, \underline{b}_t^h) \leqslant \underline{w}_{t,j}$	$\forall a \in A', \forall t \in T, \forall j \in J, \forall h \in \{0,...,k\}$	(25)
$\underline{W}_{t,j}(a, \underline{b}_t^h) \leqslant \underline{C}_{t,j}(a, \underline{b}_t^h)$	$\forall a \in A', \forall t \in T, \forall j \in J, \forall h \in \{0,...,k\}$	(26)
$\underline{w}_{t,j} + (\underline{C}_{t,j}(a, \underline{b}_t^h) - 1) \leqslant \underline{W}_{t,j}(a, \underline{b}_t^h)$	$\forall a \in A', \forall t \in T, \forall j \in J, \forall h \in \{0,...,k\}$	(27)
$\sum_{j \in J} \underline{W}_{t,j}(a, \underline{b}_t^h) \leqslant \underline{\lambda}_t + x(a, \underline{b}_t^h) - \gamma$	$\forall a \in A', \forall t \in T, \forall h \in \{0,...,k\}$	(28)
$\underline{\lambda}_t - (1 - x(a, \underline{b}_t^h)) \leqslant \sum_{j \in J} \underline{W}_{t,j}(a, \underline{b}_t^h)$	$\forall a \in A', \forall t \in T, \forall h \in \{0,...,k\}$	(29)
$\sum_{h \in \{1,...,k\}} (x(a, \bar{b}_t^h)) - \bar{b}_t^{K(a)+1} + \gamma \leqslant \bar{C}_t^+(a) \cdot k$	$\forall a \in A', \forall j \in J$	(30)
$(\bar{C}_t^+(a) - 1) \cdot k \leqslant \sum_{h \in \{1,...,k\}} (x(a, \bar{b}_t^h)) - \bar{b}_t^{K(a)+1}$	$\forall a \in A', \forall t \in T$	(31)
$\sum_{h \in \{1,...,k\}} (x(a, \bar{b}_t^h)) - \bar{b}_t^{K(a)} + \gamma \leqslant \bar{C}_t^-(a) \cdot k$	$\forall a \in A', \forall j \in J$	(32)
$(\bar{C}_t^-(a) - 1) \cdot k \leqslant \sum_{h \in \{1,...,k\}} (x(a, \bar{b}_t^h)) - \bar{b}_t^{K(a)}$	$\forall a \in A', \forall t \in T$	(33)
$\bar{W}_t^+(a) \leqslant \bar{w}_t$	$\forall a \in A', \forall t \in T$	(34)
$\bar{W}_t^+(a) \leqslant \bar{C}_t^+(a)$	$\forall a \in A', \forall t \in T$	(35)
$\bar{C}_t^+(a) + \bar{w}_t - 1 \leqslant \bar{W}_t^+(a)$	$\forall a \in A', \forall t \in T$	(36)
$\bar{W}_t^-(a) \leqslant \bar{w}_t$	$\forall a \in A', \forall t \in T$	(37)
$\bar{W}_t^-(a) \leqslant \bar{C}_t^-(a)$	$\forall a \in A', \forall t \in T$	(38)
$\bar{C}_t^-(a) + \bar{w}_t - 1 \leqslant \bar{W}_t^-(a)$	$\forall a \in A', \forall t \in T$	(39)
$\sum_{t \in T} \bar{W}_t^+(a) \leqslant \bar{\lambda} - \gamma$	$\forall a \in A'$	(40)
$\bar{\lambda} - \gamma \leqslant \sum_{t \in T} \bar{W}_t^-(a)$	$\forall a \in A'$	(41)