Analysis of the break assignment problem in emergency fleets considering area coverage

Novak, Dora

University of Zagreb
Faculty of Mechanical Engineering and Naval Architecture

ANALYSIS OF THE BREAK ASSIGNMENT PROBLEM IN EMERGENCY FLEETS CONSIDERING AREA COVERAGE

Master Thesis

Dora Novak

Zagreb, 2021

University of Zagreb
Faculty of Mechanical Engineering and Naval Architecture

ANALYSIS OF THE BREAK ASSIGNMENT PROBLEM IN EMERGENCY FLEETS CONSIDERING AREA COVERAGE

Master Thesis

Supervisors:
prof. dr. sc. Marin Lujak
(IMT Lille Douai)
prof. dr. sc. Hrvoje Cajner
(University of Zagreb)

Dora Novak

Zagreb, 2021

I want to thank to

Prof. dr. sc. Marin Lujak for providing me with a great research opportunity and mentorship,

Prof. dr. sc. Hrvoje Cajner for all the support and guidance,

My family for supporting me throughout my whole education,

And my friends, who have been by my side believing in me.

I hereby declare that I have made this thesis independently using the knowledge acquired during my studies and the cited references.

This master thesis is the result of a research project, part of a 4-month internship at IMT Lille Douai in France.

Zagreb, January 2021 Dora Novak

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske radove studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa: 602 - 04 / 21 - 6 / 1	
Ur. broj: 15 - 1703 - 21 -	

DIPLOMSKI ZADATAK

Student: **DORA NOVAK** Mat. br.: 0035204392

Naslov rada na hrvatskom jeziku: **Analiza problema dodjele vremenskih perioda odmora posadama hitnih službi uzimajući u obzir kriterij pokrivenosti područja**

Naslov rada na engleskom jeziku: **Analysis of the break assignment problem in emergency fleets considering area coverage**

Opis zadatka:

The problem of break assignment considering area coverage (BAPCAC) is a combination of the break scheduling problem (BSP) and maximum coverage location problem (MCLP) which are common operations research problems. These problems can be solved as a part of the NP-hard problems. Using historical data and statistical analysis of the spatial-temporal distribution of the tasks it is possible to achieve the optimum assignment.
The BAPCAC can also be applied to the problem of the break assignment in the emergency fleets. The workload of the emergency fleet crews usually can't be predicted with a high level of the certainty, and the urgency of the tasks needs fast reaction unconditionally. The problem arises when the crew efficiency is reduced and susceptibility to error is increased due to lack of the quality break schedule. Therefore, the break schedule of the emergency fleet crews needs to be thoroughly calculated in order to reduce fatigue and probability of the error.

The master thesis should address the following:

1. Provide the literature review and theoretical basis related to BSP and MCLP problems and describe the integration into the BAPCAC model.
2. Provide the analysis of the structure and complexity of the BAPCAC mathematical model for the example of emergency fleet problem using simulation software IBM ILOG CPLEX Studio.
3. Considering that the model has three dimensions (length of the planning time horizon, size of the space and size of the fleet) provide the correlation analysis of the dimension values and computational time.
4. Propose the simplified mathematical model for the given problem in order to obtain optimal computational characteristics of BAPCAC model. Test the simplified model using simulation experiments.

It is necessary to list the reference literature and any help received.

Zadatak zadan:	Rok predaje rada:	Predviđeni datum obrane:
12. studenog 2020.	14. siječnja 2021.	18. siječnja do 22. siječnja 2021.

Zadatak zadao:	Komentor:	Predsjednica Povjerenstva:
Izv. prof. dr. sc. Hrvoje Cajner	prof. dr. sc. Marin Lujak	prof. dr. sc. Biserka Runje

Content

List of Figures

List of Tables

List of Symbols

$\mathcal{T}$	time horizon; set of time periods in a work shift
A	set of agents representing number of capacitated vehicles with vehicle crews
B	set of break types
b_s	break type: short breaks
b_l	break type: long breaks
$\mathcal{I}$	set of unit areas with positive incident density
$\mathcal{J}$	set of unit areas, apt for vehicle locations
$D_{i\tau}$	estimated demand
Δ_b^{MIN}	minimal duration of a break
Δ_b^{MAX}	maximal duration of a break
$IM_{jj'}$	maximum allowed travel distance between two consecutive time slots matrix
N_{ij}	possibility of assistance for *idle* vehicles matrix
$\bar{N}_{ij}$	possibility of assistance for *on-break* vehicles matrix
MAX_b^w	maximum allowed work time before assigning break
$x_{a\tau j}$	binary agent allocation variable
$y_{a\tau}$	binary break assignment variable
$\alpha_{ab\tau}$	binary variable of break start
$z_{a\tau i}$	real nonnegative coverage assignment variable for *idle* vehicles
$\bar{z}_{a\tau i}$	real nonnegative coverage assignment variable for *on-break* vehicles
$\delta_{i\tau}$	real nonnegative density of area uncovered
$dev_{i\tau}$	deviation of noncoverage of area
γ	weight defining the importance of the deviation in respect to the system optimum
w	weight defining the importance of the uncoverage in respect to the objective function

List of Abbreviations

BAPCAC	Break Assignment Problem Considering Area Coverage
MILP	mixed-integer linear programming
BSP	Break Scheduling Problem
MCLP	Maximum Coverage Location Problem
OPL	Optimization Programming Language
CPU	central processing unit
RAM	random access memory

Summary

Due to accident prediction with low certainty, emergency fleet allocation becomes a challenging optimization problem. In addition, to ensure high responsiveness and efficiency of emergency crews, their work shifts need to be well balanced with enough break time. These two aspects are simultaneously taken into consideration in Break Assignment Problem Considering Area Coverage (BAPCAC) proposed by Lujak et al. in [1]. Because of its multidimensionality, when mathematically modeled, it becomes a very complex problem with high computational cost. Need for fast emergency response requires optimal results within short time. Therefore, scalability and computational complexity of the BAPCAC model need to be analyzed. After scalability analysis, heuristic simplification of the mathematical model is proposed, so that BAPCAC problem can be solved within reasonable time.

Keywords: break scheduling, maximal area coverage, mixed-integer linear programming, emergency fleet scheduling, computational complexity

1 Introduction

Efficiency of the emergency service fleets can be greatly impacted by the level of fatigue, with their responsiveness being lower than expected if crews composing their fleets did not have sufficient rest. They have to deal with stressful situations during the whole work shift and be ready to assist urgent accidents at any given moment. It is, therefore, important to ensure high alertness level by ensuring enough breaks in terms of length and occurrence.

Work load cannot be predicted with certainty because of the stochastic appearance of accidents both in terms of location and time. This means that number of vehicles needed to assist the incidents is unknown and can only be predicted by applying statistical methods on historical data. Minimum number of emergency vehicle crews should be available in each time period of the planning time horizon and for certain area, even though they might not be constantly needed for assistance. Also, incidents are often urgent and need to be assisted within certain time, e.g. ambulance should reach the incident location within 20 min from the call. Chances of saving lives are higher if response is faster.

Novel optimization problem is formulated as Break Assignment Problem Considering Area Coverage (BAPCAC) [2][3], which combines two previously investigated operations research problems: break scheduling and maximization of area coverage. BAPCAC mathematical model aims to minimize incident area that remains uncovered by vehicles, while minimizing fatigue of agents employed. Aim of the model is to position agents so that they are able to assist given incidents but also to ensure enough rest time so that their responsiveness is on a required level. Emergency service fleet should be sized according to the given estimated demand, which is a statistically obtained input for the model.

BAPCAC is a mixed-integer linear programming (MILP) problem and, as such, highly computationally complex. To ensure optimal solutions within reasonable computational time, scalability of the problem needs to be investigated. Since BAPCAC is a spatio-temporal problem that finds work breaks and the positions in each time period of a work shift for emergency vehicles in a region of interest, its complexity can be explored in 3 dimensions: time, space, and the size of the crew. The dimension that contributes to the model computational cost at the highest level should be revised and simplification should be proposed in order to obtain good coverage of the region of interest by emergency vehicles within needed time without compromising the crews' breaks required for an efficient and effective operation.

2 Preliminaries and related work

This research work is focused on break assignment problem considering area coverage (BAPCAC) as a combination of two separate problems: break scheduling problem (BSP) [4] and the maximum coverage location problem (MCLP) [5]. Both problems have already been explored as research topics in operations research. Most relevant studies for both BSP and MCLP will be discussed in this chapter and later used as fundamentals for developing mathematical formulation for BAPCAC model [1].

2.1. Break scheduling problem

Definition of the break scheduling problem given in [4] refers to an optimization problem of break assignment in a defined time horizon, while various constraints are satisfied. BSP is proven to be a NP-complete problem and therefore highly complex.

Research papers focus on scheduling problem in different areas of application. Robbins and Harrison in [6] focus on scheduling in call centers while handling high uncertainty level and volatility of call volumes. Fei et al. in [7] study scheduling surgeries in an operating theatre, so that utilization of operating rooms is optimized, while idle time between surgeries is minimized. Aickelina and Dowslandb in [8] are solving the problem of scheduling shifts for nurses with respect to working contracts, meeting the demand and evenly distributing nurses.

Di Gaspero et al. in [9] tackle the shift design problem as a variant of the shift scheduling problem. Authors generate set of shifts, which are completely defined in terms of their duration and start time. There are seven criteria imposed as soft constraints including break and work load constraints, such that there is neither shortage nor excess of employees for each time slot.

Reikk et al. in [10] investigate flexibility in shift scheduling problem considering shift starting times, break duration and placement. They propose fractional breaks, that do not need to be assigned as a whole, and work stretch duration restrictions, that would ensure optimal combination of work and rest in a single work shift. Positive impact of fractional breaks on reducing workforce size is analyzed, as well as improvement of schedule quality when work stretch duration restrictions are applied.

Quimper and Rousseau in [11] propose a Large Neighbourhood Search approach for modeling restrictions within scheduling problem with fluctuating staff demand. This method ensures near-optimal solution by ensuring larger scale problem solving.

2.2. Maximal area coverage problem

Church and Revelle in [5] describe the fundamentals of maximal covering location problem (MCLP). It can be set as either maximization or minimization problem, if it is solved by applying linear programming. The aim is to allocate available resources so that demand or area remained uncovered is minimal. Authors indicate two cases considered in linear programming approach if total coverage is not possible: "all-integer answer" that appears in 80% of all problems or "fractional answer" that appears only in 20% of time. In order to transform fractional into all-integer solutions authors recommend two algorithms, method of inspection and method of Branch and Bound.

Area coverage often addresses problems of wireless sensor networks. Area coverage in this case is rather probabilistic so the problem is difficult to solve. Gallais and Carle in [12] focus on optimization of node efficiency in a wireless sensor network in terms of minimization of energy consumption by activating only the necessary nodes but with maximum coverage possible.

Brotcorne et al. in [13] analize ambulance already existing location and relocation models. They evaluate static models for this problem and analyze attempts of solving the same problem dynamically. Further examples of relocation models for ambulance are explored in [14][15][16][17].

3 Break assignment problem considering area coverage (BAPCAC)

3.1. Overview of the BAPCAC problem

Break assignment problem considering area coverage (BAPCAC) is a first attempt of simultaneously solving two problems: the break scheduling problem (BSP) and the maximum coverage location problem (MCLP). These two problems, both NP-hard, were previously considered as separate issues and none of the modelling approaches has taken into account their compatibility and necessity of a model that would integrate and solve both of them at once as a novel NP-hard problem. For the self-sufficiency of this work, the BAPCAC model that was previously developed by Lujak et al. in [2][3] is presented in this thesis.

The BSP problem aims to find the optimal break schedule within systems that have a known and definite number of vehicle crews available to be assigned to certain tasks. Break scheduling can be of a great importance for numerous systems, such as police patrols or out-of-hospital emergency medical assistance, where it is necessary to have a shift with enough rest time, in order to keep the system efficiency at the highest possible level. In BSP problem shifts and various types of breaks are planned by satisfying different constraints, legal and other, while avoiding or minimizing fatigue.

On the other hand, the MCLP problem focuses on minimizing the area that is not covered by available units. The goal is to distribute these units according to the spatial distribution of demand with a temporal dimension. The algorithm should be able to dynamically distribute the units, while closing the gap between demand and supply, which means that it would relocate the units in real time, depending on how the demand distribution changes.

However, when solving these two problems separately and sequentially, parts of the issue crucial for optimal functioning of systems are omitted. This is why it is important not to exclude any goal of the two problems and solve them in a mathematical program that unites them both. Therefore, a new break assignment problem considering area coverage (BAPCAC) is formulated as a combination of BSP and MCLP. BAPCAC is taking into consideration both spatial and temporal dimension of demand as the main input, and assign breaks and distribute vehicles over the region of interest according to this demand.

3.2. Challenges of the problem

BAPCAC should be able to provide an optimal solution for two rather contradictory problems at the same time: minimizing the uncovered area, while minimizing the workload for work units, or in this case emergency vehicles. The contradiction can be found in the fact that with less vehicles available there would be more area that cannot be assigned to any of the idle vehicles, thus more uncovered area.

Considering that the preferences of most of the emergency crews for meal breaks (breakfast, lunch, dinner) overlap in time windows, that is also when the most of the area remains uncovered, with not enough idle vehicles available to assist the occurring incidents. When focused only on break scheduling, models do not take into account area coverage, which implies less system efficiency in certain time periods. To avoid this issue, it is necessary to schedule breaks in that way that all the emergency crews get enough rest time, while ensuring demand coverage even during lunch time. Coverage should remain priority when deciding upon optimal break scheduling.

One of the challenges is represented by the size of the area that one unit is able to cover in one time period. For out-of-hospital emergency medical assistance, i.e. emergency vehicles, it is crucial that they can reach and assist an incident within 15-20 minutes. When approaching this optimization problem, it is needed to decide upon the size of an area unit, while taking into account all the infrastructural conditions that affect the traveling time. The area units should be sized in that way that vehicles can reach any point in the surrounding area units within the required time. This prerequisite would ensure that the incidents are assisted appropriately. Number of area unites grows with scaling down their size. With larger number of area units BAPCAC problem becomes more complex and its solving turns out to be more expensive. On the other hand, if a larger area is considered as one unit, there is a greater error in knowing the exact demand distribution across this larger area. Therefore, it is necessary to find a good compromise between solving complexity and finding the optimal solution.

Another purpose, thus a challenge, of the BAPCAC problem is to ensure scalability. In other terms, the mathematical formulation should result in a model that is suitable to solve the same problem in different environments, given different parameters. Problem scalability should support changing and growing number of variables, especially the number of area units, available vehicles and time periods. Bigger number of variables introduced to an optimization model will cause higher computational complexity, and consequently longer computational time. Given the nature of this problem, BAPCAC model should be able to give optimal solution in real time. If the optimal solution is found, but the calculation time is unacceptably long, that model is not convenient for BAPCAC problem.

3.3. Mathematical formulation of the BAPCAC problem

In order to model the problem, parameters and variables have to be declared. They are given in the table below, followed by the BAPCAC mathematical program. The goal is to optimize the mathematical program. This means, within all the feasible solutions, find the one that maximizes or minimizes the objective function.

3.3.1. Mathematical programming model

Sets and indices

$\mathcal{T}$	time horizon; a set of time periods in a work shift; $\tau \in \mathcal{T}$
A	set of agents $a \in A$ representing $\|A\|$ capacitated vehicles with vehicle crews
B	set of break types for example, $\{b_s, b_l\}$, where b_s stands for short breaks, whereas b_l stands for long breaks
$\mathcal{I}$	set of unit areas $i \in \mathcal{I}$ with positive incident density
$\mathcal{J}$	set of unit areas $j \in \mathcal{J}$, apt for vehicle locations

Parameters

$D_{i\tau}$	estimated demand and, thus, required minimum proportion of vehicles for unit area $i \in \mathcal{I}$ at time $\tau \in \mathcal{T}$
Δ_b^{MIN}	minimal duration of break $b \in B$
Δ_b^{MAX}	maximal duration of break $b \in B$
$IM_{jj'}$	valued 1 if an agent can move from $j \in \mathcal{J}$ to $j' \in \mathcal{J}$ based on the maximum allowed travel distance between two consecutive time slots; 0 otherwise
N_{ij}	1 if an idle vehicle in $j \in \mathcal{J}$ can assist an incident at $i \in \mathcal{I}$ within a given target arrival time; 0 otherwise
$\bar{N}_{ij}$	1 if a vehicle on a break located at $j \in \mathcal{J}$ can assist an incident at $i \in \mathcal{I}$ within a given target arrival time; 0 otherwise
MAX_b^w	maximum allowed work time before assigning break $b \in B$. Usually, short breaks each 3 h, and large ones each 4-6 h

Decision variables

Location variables

$x_{a\tau j}$	valued 1 if agent a at time $\tau \in \mathcal{T}$ is located at unit area $j \in \mathcal{J}$; 0 otherwise

Breaks variables

$y_{a\tau}$	binary break assignment variable valued 1 if agent $a \in A$ is assigned a break at time $\tau \in \mathcal{T}$; 0 otherwise
$\alpha_{ab\tau}$	binary variable valued 1 if agent $a \in \mathcal{A}$ starts a break of type $b \in B$ at $\tau \in \mathcal{T}$; 0 otherwise

Coverage variables

$z_{a\tau i}$	real nonnegative coverage assignment variable representing the part of density at incident unit area $i \in \mathcal{I}$ assigned to *idle* agent $a \in A$ at time $\tau \in \mathcal{T}$, where $0 \leq z_{a\tau i} \leq 1$
$\bar{z}_{a\tau i}$	real nonnegative coverage assignment variable representing the part of density at incident unit area $i \in \mathcal{I}$ assigned to *on-break* agent $a \in A$ at time $\tau \in \mathcal{T}$, where $0 \leq \bar{z}_{a\tau i} \leq 1$
$\delta_{i\tau}$	real nonnegative density of area $i \in \mathcal{I}$ uncovered at time $\tau \in \mathcal{T}$
$dev_{i\tau}$	deviation of noncoverage of area $i \in \mathcal{I}$ at time $\tau \in \mathcal{T}$

(BAPCAC):

$$\min \; w \cdot \sum_{i \in \mathcal{I}, \tau \in \mathcal{T}} \Big((1-\gamma)\delta_{i\tau} + \gamma dev_{i\tau} \Big) + (1-w) \cdot \sum_{a \in A, \tau \in \mathcal{T}} \big(1 - y_{a\tau}\big), \tag{3.1}$$

such that:

$$dev_{i\tau} \geq \bar{\delta} - \delta_{i\tau}, \forall i \in \mathcal{I}, \tau \in \mathcal{T} \tag{3.2}$$

$$dev_{i\tau} \geq \delta_{i\tau} - \bar{\delta}, \forall i \in \mathcal{I}, \tau \in \mathcal{T} \tag{3.3}$$

$$\sum_{a \in A} \big(z_{a\tau i} + \bar{z}_{a\tau i} \big) = D_{i\tau} - \delta_{i\tau}, \forall i \in \mathcal{I}, \tau \in \mathcal{T} \tag{3.4}$$

$$z_{a\tau i} \leq \sum_{j \in \mathcal{J} | N_{ij} = 1} D_{i\tau} x_{a\tau j}, \forall a \in A, \tau \in \mathcal{T}, i \in \mathcal{I} \tag{3.5}$$

$$\bar{z}_{a\tau i} \leq \sum_{j \in \mathcal{J} | \bar{N}_{ij} = 1} D_{i\tau} x_{a\tau j}, \forall a \in A, \tau \in \mathcal{T}, i \in \mathcal{I} \tag{3.6}$$

$$z_{a\tau i} \leq D_{i\tau}(1 - y_{a,\tau}), \forall a \in A, \tau \in \mathcal{T}, i \in \mathcal{I} \tag{3.7}$$

$$\bar{z}_{a\tau i} \leq D_{i\tau} y_{a,\tau}, \forall a \in A, \tau \in \mathcal{T}, i \in \mathcal{I} \tag{3.8}$$

$$\sum_{i \in \mathcal{I}} z_{a\tau i} \leq 1, \forall \tau \in \mathcal{T}, a \in A \tag{3.9}$$

$$\sum_{i \in \mathcal{I}} \bar{z}_{a\tau i} \leq 1, \forall \tau \in \mathcal{T}, a \in A \tag{3.10}$$

$$\sum_{j \in \mathcal{J}} x_{a\tau j} = 1, \forall \tau \in \mathcal{T}, a \in A \tag{3.11}$$

$$x_{a\tau j} + x_{a(\tau+1)j'} \leq 1 + IM_{jj'}, \forall a \in A, \tau \in [1, |\mathcal{T}| - 1], j, j' \in \mathcal{J} \tag{3.12}$$

$$y_{a\tau} \geq \sum_{b \in B} \sum_{k=0}^{\Delta_b^{MIN}-1} \alpha_{ab(\tau-k)}, \ \forall a \in A, \ \tau \in [\max_{b \in B}(\Delta_b^{MIN}), |\mathcal{T}|] \tag{3.13}$$

$$y_{a\tau} \leq \sum_{b \in B} \sum_{k=0}^{\Delta_b^{MAX}-1} \alpha_{ab(\tau-k)}, \ \forall a \in A, \ \tau \in [\max_{b \in B}(\Delta_b^{MAX}), |\mathcal{T}|] \tag{3.14}$$

$$\tag{3.15}$$

$$\sum_{\tau'=\tau}^{\tau+MAX_b^w-1} \alpha_{ab\tau'} \geq 1, \ \forall a \in A, b \in B, \tau \in [1, |\mathcal{T}| - MAX_b^w + 1] \tag{3.16}$$

$$\sum_{b \in B} \alpha_{ab\tau} \leq y_{\alpha\tau}, \ \forall a \in A, b \in B, \tau \in [1, |\mathcal{T}|] \tag{3.17}$$

$$z_{a\tau i}, \bar{z}_{a\tau i}, \delta_{i\tau} \geq 0, x_{a\tau j}, y_{a\tau}, \alpha_{ab\tau} \in \{0, 1\}, \forall i \in \mathcal{I}, j \in \mathcal{J}, \tau \in \mathcal{T}, a \in A \tag{3.18}$$

BAPCAC problem consists of three dimensions: temporal, spatial and demand dimension. That makes this problem rather complex, especially when the number of variables in those dimensions is growing. In addition to setting the right objective function, numerous of constraints are necessary so that connection and interaction between the parameters and decision variables are enabled. Furthermore, constraints ensure that the optimal solution is at the same time feasible. A feasible solutions should satisfy all the requirements and limitations of a real system that is being mathematically modeled.

BAPCAC, as mentioned above, is simultaneously solving two problems, and therefore, each of the two problems will be assigned with certain weight w, which will enable to choose different focus for different environments. For instance, delivery service might need to prioritize breaks over the area coverage, while for emergency fleet it would be the opposite. The objective function (3.1) is consisted of two parts - the first part that minimizes uncovered area, and the second part that is maximizing the number of breaks assigned to the agents. Those two functions are opposed to each other, and therefore, weight w, such that $0 \leq w \leq 1$, is added to the function, which allows that weighted importance is assigned to each of the two parts. Objective function does not necessarily need the second part, but it can ensure greater efficiency of the agents by decreasing the fatigue. However, because maximizing the *on-break* time, it necessarily decreases the number of the *idle* vehicles available to assist the incidence, thus, as a result increases the uncovered area, i.e. interferes with the first part of the objective function. This problem could be solved by ensuring the demanded break time only with constraints. That way the part of minimizing the fatigue would be omitted, but the problem could be simplified.

Expressions (3.2) and (3.3) are introducing deviation parameter $dev_{i\tau}$. The optimal solution of the objective function is being found with minimizing the value of the objective function. Since this function (3.1) considers only the summation of all the values over all time periods and all geographical areas (cells), it does not consider the performance of individual cells and individual time periods. For example, considering 2 cells or 2 time periods, a sum of 10 can be 1+9, 2+8, 3+7, etc. Even though 10 may be the optimal value, the sum does not consider how this value is achieved within the sum. Thus, it should be as balanced as possible, i.e., ideally 5+5, and to have that guaranteed, deviation is introduced. The deviation should be minimized. Related weight γ, such that $0 \leq \gamma \leq 1$, defines the importance of the deviation in respect to the optimal solution or the system optimum. The deviation is formulated as follows:

$$dev_{i\tau} = |\bar{\delta} - \delta_{i\tau}|, \forall i \in \mathcal{I}, \tau \in \mathcal{T} \tag{3.19}$$

For the purpose of linear programming, non-linear expression (3.19) needs to be reformulated into linear expressions (3.2) and (3.3). There, $\bar{\delta}$ is defined as:

$$\bar{\delta} = \frac{\sum_{i \in \mathcal{I}, \tau \in \mathcal{T}} \delta_{i\tau}}{|\mathcal{T}||\mathcal{I}|}, \tag{3.20}$$

where $|\mathcal{T}|$ is cardinality of $\mathcal{T}$ and $|\mathcal{I}|$ is cardinality of $\mathcal{I}$.

Constraint (3.4) links the covered area part with *idle* and *on-break* vehicles. The sum of all density parts assigned to the *idle* and *on-break* vehicles in each area and time slot should amount to a maximum of the density of covered area. The coverage variable $\bar{z}_{a\tau i}$ in the constraint assumes that the breaks are preemptive and can be ended if an incident needs to be assisted by the *on-break* vehicle. If this variable is excluded from the constraints, the model will turn into a non-preemptive break model.

Constraints (3.5) and (3.6) limit the coverage of each idle agent and an agent on-break to at most incident density $D_{i\tau}$ of unit area $i \in \mathcal{I}$ at time slot $\tau \in \mathcal{T}$ if it is positioned within the travel time defined by adjacency matrix N_{ij} and $\bar{N}_{ij}$, respectively. These constraints ensure that part $z_{a\tau i}$ and of the coverage of the area i by an idle agent and an agent on-break, respectively, is at most the sum of the densities of the areas $j \in \mathcal{J}$ within its reach at time $\tau \in \mathcal{T}$. Furthermore, constraints (3.7) and (3.8) limit idle coverage and the coverage on-break vehicles only to idle agents and agents at break, respectively. Moreover, constraint (3.9) guarantee that the incident density covered by each idle agent sums up to at most 1 at time $\tau \in \mathcal{T}$ due to the unitary vehicle capacity, i.e. each agent can assist only one incident at once. Similarly, constraint (3.10) limits the capacity for the coverage of each agent on-break.

Constraint (3.11) assigns to each agent a at each time slot τ a unique location $j \in \mathcal{J}$, by means of location variable $x_{a\tau j}$, while constraint (3.12) constraints agents' positions $x_{a\tau j}$ in two consecutive time slots t, $t+1$ to be within the maximum allowed distance travelled established by parameter $IM_{jj'}$ with binary values.

Minimal and maximal break durations are imposed by constraints (3.13) and (3.14). Constraint (3.15) prohibit the assignment of more than one break type during a maximal duration of each break. For a time window starting at each time slot with the duration of a maximum uninterrupted work time MAX^w_b, each break type should start at least once according to (3.16). An alternative to this inequality is the equal sign where no matter the coverage requirements, vehicle crews take exactly one break of each type in the maximum allowed work time even though their presence is not necessary for keeping the coverage.

Moreover, constraint (3.17) prohibits the start of more than one break type at a time, while constraint (3.18) represents non-negativity and range constraints on decision variables.

Regarding the parameters, all of them except for the estimated demand are known and deterministic values. When it comes to demand, only an estimation is given. It corresponds to a proportion of patients that is in certain area. Since one vehicle is assigned to one patient in the real allocation of patients to vehicles, the number of patients will then correspond to the number of vehicles. Demand parameter is probabilistic and is based on historically gathered data. Statistical models with autoregressive and seasonality components, e.g. spatial autoregressive moving average (SARMA), can be used to obtain the demand density input parameter.

The error size of demand estimation depends on the size of the unit areas and duration of time periods. If the unit area size or the duration of one time period is rather small, the associated error will be proportionally small, which means that demand estimation would be more accurate for that specific time slot in certain area unit. However, if the whole area of the given city has to be considered, by scaling down one area unit size, the number of the units would lead to an increasing number of variables which would eventually affect the computational time and complexity. Computational complexity and time are often the issue with the exact solving methods. It is important to see how those two change by simulating different instances with different number of variables applied to the same model. To get the solution faster, heuristic methods can be applied instead of exact methods. However, heuristic methods do not guarantee the quality of given solution. Quality guarantee in exact methods is achieved by gap, i.e. the parameter that shows what is the difference between the given and optimal solution after certain computational time has passed. Therefore, a balance between the accuracy, optimum and the computational time should be found.

Also, the arrival of the emergency crew within the requested time of 15-20 minutes is modeled by the area unit size in the way that each vehicle can access any point of the surrounding units from any point of the initial unit within the allowed time. As the size of the area units is being changed, this request is no longer fulfilled by the same model consisted of the same constraints.

The optimal solution to BAPCAC problem is supposed to be found in the rolling time horizon, thus for the next $|\mathcal{T}|$ number of periods before each period starts. As opposed to finding the optimal schedule at the beginning of the work shift for its complete duration, this should provide real-time optimization. To be able to perform the optimization fast enough, mathematical complexity of the model should be reduced. For this reason, an offline optimization will be performed first to see how the model behaves with all information assumed to be deterministic and known a priori. Only after this step the focus will be moved on to the probabilistic version of it.

This is a hardly NP-hard problem since both break assignment problem and maximum area coverage problem are very computationally complex problems.

BAPCAC problem is focusing only on scheduling breaks for emergency crews while distributing the crews optimally in the considered area. Mathematical model does not distinguish whether an *idle* or *on-break* vehicle should assist the occurring incident. As a next step, incident assignment should take into account the preference of assigning the incident to an idle vehicle, in order to avoid break disruptions.

3.3.2. Mixed-integer linear programming problem (MILP)

BAPCAC optimization problem is defined as a mixed-integer linear programming problem (MILP) because its variables are both integer and continuous. By definition, it is a non-convex problem, which contributes to its high complexity. Branch-and-bound algorithm is often used for solving MILP optimization problems with linear relaxation (see eg., [18]). In general, MILP problems are NP-hard. When reduced to BSP, BAPCAC is proven to be NP-complete (see eg., [19]).

Branch-and-bound algorithm divides the initial problem into subproblems, which are arranged in tree structure. Algorithm is searching for solutions in these tree branches, where it excludes solution branches that are either infeasible or no better than previously found solutions. Each subproblem creates lower and upper bound of solutions, and aims for either lower bound for minimization problem or upper bound for maximization problem (see eg., [20])

4 Model output simulation results

In order to later test the scalability and all the limitations of the model, it is necessary to ensure the logic between the input and the output of the model. With having in mind the expected output, certain hypotheses will be tested.

These simulations will be performed with IBM ILOG CPLEX Optimization Studio version 12.10, a decision optimization software based on prescriptive analytics using mathematical or constraint programming. CPLEX is a high-performance software that supports Optimization Programming Language (OPL) [21]. CPLEX solver uses methods based on branch-and-bound algorithm to propose optimal solution for BAPCAC problem, interpreted as a MILP problem.

As the main goal is to determine computational complexity of the model, which directly influence computational time, it is only possible to compare the results in relative terms. Therefore, all the simulation experiments will be run on the same computer, Dell laptop with with Intel Core i7-7560U CPU running at 2.40Ghz and with 16Gb of RAM. Upper time limit for all the simulations will be 3600 seconds.

4.1. Experiment setup

4.1.1. Emergency fleet use case example model

To graphically describe the model, an example of a simple system will be explained. This system will include 2 vehicles that need to be placed in 16 area cells (area 4x4) (Figure 4.1) so that minimal incident density remains uncovered, while minimizing fatigue of their crew. Uncoverage is modeled by the first part and fatigue by the second part of the objective function (3.1). Incident density for each time slot $\tau \in \mathcal{T}$ and each area cell $i \in \mathcal{I}$ is represented by the estimated demand or the probability that the incident will need assistance. Estimated demand $D_{i\tau}$ is therefore a real number. Each of the vehicles in number of vehicles $NbVeh$, equivalent to $|\mathcal{A}|$ in chapter 3., can cover $D_{i\tau} \leq 1$ incidents. *Idle* vehicles can assist all the incidents that are happening right, left, up, down from the area cell where the vehicle is currently placed, as well as the diagonals and the area cell of the vehicle current position. Considering need for additional time to end the break and start the assistance, *on-break* vehicles will not be able to cover the area cells diagonally from their current position. This constraint is modeled by N_{ij} adjacency matrix in (3.5) and $\bar{N}_{ij}$ in (3.6) for *idle* and *on-break* vehicles, respectively. Each vehicle can cover between 3 and 9 area cells, depending on its current position.

4.1.2. Instance example for emergency fleets

For example, two-vehicle fleet in Figure 4.1 represents the time slot t, where black vehicle is *idle*, placed at B2 area cell that can possibly assist 9 area cells colored in blue. On the other hand, grey vehicle is currently *on-break* and therefore able to cover only 4 area cells. Area cell C3 is colored in darker shade of blue because both vehicles can assist the incidents happening there.

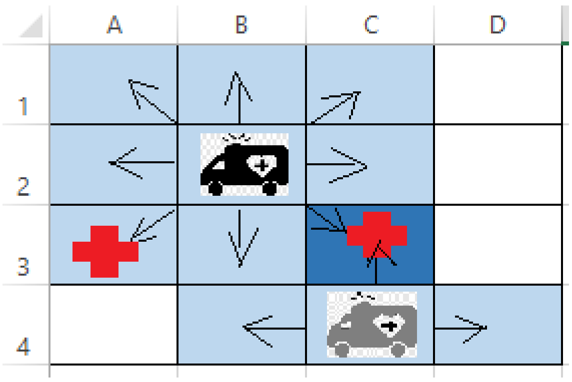

Figure 4.1: Simple system with region of interest with 16 cells (area 4x4) and two vehicles; time slot t

Incident density, i.e. demand estimation for time slot t is given in Table 4.1. If vehicles are placed as in Figure 4.1, black vehicle is able to cover A3 demand $D_{i\tau} = 0.5$, and C3 demand $D_{i\tau} = 0.5$. At the same time, grey vehicle will assist the rest of the C3 demand, $D_{i\tau} = 1$. Therefore, no demand would remain uncovered for time slot t.

Area cell	Demand estimation
A3	0.5
C3	1.5

Table 4.1: Demand estimation for time slot t

For the consecutive time slot $t + 1$, it is important understand and limit movement possibilities of the vehicles. In consecutive time slots, vehicles can move in all the directions for one area cell: up, down, left, right and on diagonals. This movement is defined by adjacency matrix $IM_{jj'}$ in the constraint (3.12). For time slot $t + 1$, both vehicles are *idle* (Figure 4.2) and can assist up to 9 area cells. Demand estimation is given in Table 4.2. Vehicle currently placed in C2 will therefore optimally assist $D_{i\tau} = 1$ of the incident density in D1, which means that the rest of its density $D_{i\tau} = 0.25$ would remain uncovered. Incident density in A4 will be fully assisted by the vehicle placed in B4.

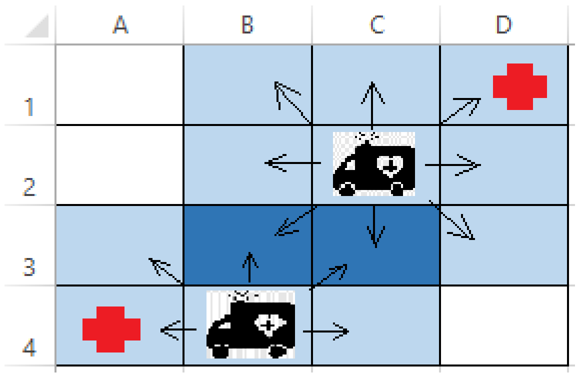

Figure 4.2: Simple system with region of interest with 16 cells (area 4x4) and two vehicles; time slot $t + 1$

Area cell	Demand estimation
A4	0.75
D1	1.25

Table 4.2: Demand estimation for time slot $t + 1$

For both time slots, t and $t + 1$, incident density is normalised according to the number of available vehicles, meaning $\Sigma D_{i\tau} = 2$ distributed over all 16 area cells for each time slot.

Furthermore, need for breaks has to be mathematically modelled. Two break types are defined for further simulation experiments, long and short breaks. Parameters for each type are shown in Table , where duration of one time slot t is 20 minutes.

	Short break	Long break
Δ_b^{MIN}	20’	40’
Δ_b^{MAX}	20’	1h20’
MAX_b^w	1h40’	3h20’

Table 4.3: Break parameters for simulation experiments

4.1.3. Initial simplifications

Because of the high-complexity nature of the MILP problem, if number of variables and constraints is increased, computational complexity can grow exponentially. Hence, some initial simplifications will be implemented to the BAPCAC mathematical model, so that CPLEX can more easily find the initial solutions within a reasonable computational time. Two simplifications will be considered. First, deviation parameter $dev_{i\tau}$ from (3.1) will be removed, which will imply elimination of constraints (3.2) and (3.3), which create exponential increase in number of parameters by linearizing an initially non-linear variable $dev_{i\tau}$. By removing deviation from the model, $|\mathcal{I}|\text{x}|\mathcal{T}|$ number of constraints would be eliminated twice, one time for each constraint removed. Second simplification will consider elimination of the variable $\bar{z}_{a\tau i}$, concerning assignment of incidents to *on-break* vehicles. Thus, constraint (3.6), (3.10) will be excluded, while constraint (3.4) will be simplified when only considering assigning incidents to the *idle* vehicles, i.e. only variable $z_{a\tau i}$. Without constraints (3.6) and (3.10), $|\mathcal{I}|$x$|\mathcal{T}|$x$|\mathcal{A}|$ and $|\mathcal{T}|$x$|\mathcal{A}|$ number of constraints will be removed respectively. These two simplifications will be implemented for all simulation experiments and analysis of the BAPCAC model.

4.2. Simulation analysis

4.2.1. Simulation analysis within the same instance

For each simulation instance there are three dimensions that can be changed: number of vehicles, number of time periods and area size. To verify that the model gives the expected results, the simple instance with 4 vehicles, 12 time periods and with region of interest with 16 cells (area 4x4) will be simulated. Also, to decrease the computational complexity, model is simplified and thus deviation and assigning incident density to vehicles *on-break* is excluded. Results are given in the Table 4.4. For the demand variable, incident density is normalised for 4 vehicles, which means that supply and demand are the same, i.e. in each time period and area cell there are in total 4 accidents that need assistance of 4 vehicles. For the same instance, different weight w will be assigned each time, varying from 0 to 1, with a change of 0.1.

Assigning different weight w values to the model will give more or less importance to each side of the objective function. Results of the simulation in the Table 4.4. show that uncoverage is increasing as the value of w is decreasing, and fatigue is decreasing together with the w, as expected. Figure 4.3 represents graphically change in uncoverage and fatigue with decreasing value of the weight w. Values of uncoverage and fatigue remain rather constant for all the w values between 0.6 and 1. There is a big jump around $w = 0.5$. For the w values between 0 and 0.4 take their maximal, i.e. minimal values. Even though the direction of both lines is as expected, smoother transition would be desirable. Ideally, uncoverage values would be gradually decreasing together with the weight w, while fatigue would increase with the same intensity. For this particular instance, at lower values of the weight w model assigns all the importance to fatigue variable, without differentiating and distributing the importance among the two decision variables. When $w < 0.5$, model assigns break to all the *idle* vehicles for all the 12 time

No. of Agents	Weight w	Deviation	Zbar	Gap [%]	Uncoverage	Fatigue	Real time [sec]
4	1	NO	No	0.00	16	32	0.69
4	0.9	NO	No	0.00	16	32	0.41
4	0.8	NO	No	0.00	16	32	0.44
4	0.7	NO	No	0.00	16	32	0.36
4	0.6	NO	No	0.00	16	32	0.56
4	0.5	NO	No	0.00	44	4	0.34
4	0.4	NO	No	0.00	48	0	0.28
4	0.3	NO	No	0.00	48	0	0.22
4	0.2	NO	No	0.00	48	0	0.09
4	0.1	NO	No	0.00	48	0	0.06
4	0	NO	No	0.00	48	0	0.05

Table 4.4: Region of interest with 16 cells (area 4x4); Tmax=12; Demand normalised according to the No. of Agents

periods. Obtained results might be different if some additional constraints were added, such as to limit the time spent on break for each vehicle. Table 4.4 also shows decrease in computational time with lower values of w. This implies lower complexity of the second part of the objective function that optimizes fatigue of emergeny fleets.

4.2.2. Simulation analysis between different instances

In order to acquire additional knowledge about the model and its complexity, simulate different instances will be simulated while keeping certain dimensions constant. The results will be analyzed on a relative level among instances in order to draw conclusions about the model scalability. To better understand how uncoverage and fatigue are influenced by the model, number of available vehicles will be changed, and all the other variables will remain constant. Regarding the incident density, two situations will be considered: first, all the densities will be normalised according to the number of vehicles; and second one will consider constant incident density normalised for the

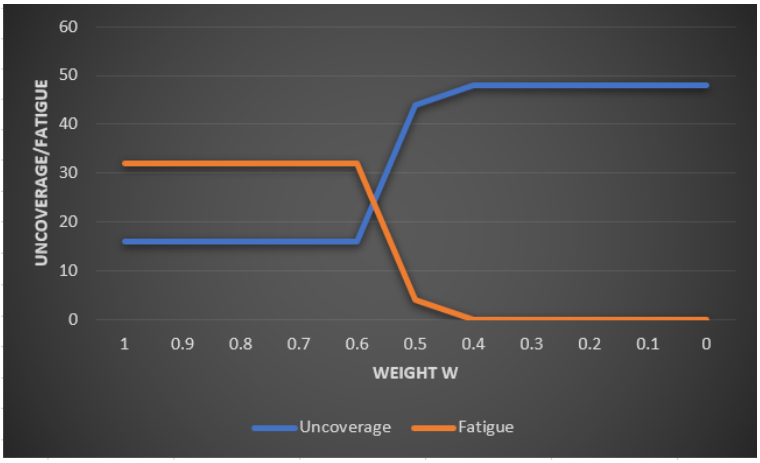

Figure 4.3: Changes in uncoverage and fatigue while decreasing weight w

same number of vehicles. Incident density will therefore directly influence the results, especially decision variable uncoverage.

Simulation analysis with incident density normalised according to the number of vehicles

The first analysis is performed as the incident density is changed and normalised for the input number of vehicles. The goal is to test how different variables are going to change when simulating several instances. Results for 4, 5 and 6 vehicles available for assistance are shown in the Figure 4.4, while instance results for 7, 8, 9 and 10 vehicles are shown in the Figure 4.5. The simulation results in Figures 4.4 and 4.5 show that total uncoverage and fatigue increase with the increasing number of vehicles when density is adjusted accordingly. However, when these variables are divided by the number of agents, values of uncoverage per agent and fatigue per agent remain rather constant. These results are projection of the mathematical model and the form of the objective function itself. The two parts of the objective function (3.1), both uncoverage and fatigue, are formed as sum of all the results for the total number of periods, number of agents and number of area cells. Therefore, to better understand the changes between different instances, it is needed to consider each value divided by the number of the agents, i.e. vehicles. The results show that each instance, with the exception of when $NbVeh = 6$, have the $Uncoverage/Agent = 4$ and $Fatigue/Agent = 8$ when all the importance is given to the uncoverage part of the objective function. Here, the ideal case is considered, because the number of vehicles available is supposed to be sufficient to cover all the incident density if no vehicle is *on-break*.

Area 4x4; Tmax=12; Demand normalised according to the No. of Agents

No. of Agents	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
4	1	0	NO	No	16	16	0.00	16	32	0.69	4	8
4	0.9	0	NO	No	17.6	17.6	0.00	16	32	0.41	4	8
4	0.8	0	NO	No	19.2	19.2	0.00	16	32	0.44	4	8
4	0.7	0	NO	No	20.8	20.8	0.00	16	32	0.36	4	8
4	0.6	0	NO	No	22.4	22.4	0.00	16	32	0.56	4	8
4	0.5	0	NO	No	/	24	0.00	44	4	0.34	11	1
4	0.4	0	NO	No	/	19.2	0.00	48	0	0.28	12	0
4	0.3	0	NO	No	/	14.4	0.00	48	0	0.22	12	0
4	0.2	0	NO	No	9.6	9.6	0.00	48	0	0.09	12	0
4	0.1	0	NO	No	4.8	4.8	0.00	48	0	0.06	12	0
4	0	0	NO	No	/	/	0.00	48	0	0.05	12	0
									AVG time	0.32		

Area 4x4; Tmax=12; Demand normalised according to the No. of Agents

No. of Agents	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
5	1	0	NO	No	20	20	0.00	20	40	0.91	4	8
5	0.9	0	NO	No	22	22	0.00	20	40	0.88	4	8
5	0.8	0	NO	No	24	24	0.00	20	40	0.75	4	8
5	0.7	0	NO	No	26	26	0.00	20	40	0.69	4	8
5	0.6	0	NO	No	28	28	0.00	20	40	0.66	4	8
5	0.5	0	NO	No	/	30	0.00	60	0	0.45	12	0
5	0.4	0	NO	No	/	24	0.00	60	0	0.38	12	0
5	0.3	0	NO	No	/	18	0.00	60	0	0.27	12	0
5	0.2	0	NO	No	/	12	0.00	60	0	0.31	12	0
5	0.1	0	NO	No	6	6	0.00	60	0	0.08	12	0
5	0	0	NO	No	/	/	0.00	60	0	0.05	12	0
									AVG time	0.49		

Area 4x4; Tmax=12; Demand normalised according to the No. of Agents

No. of Agents	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
6	1	0	NO	No	14	14	0.00	14	46	1.23	2.33	7.67
6	0.9	0	NO	No	17.2	17.2	0.00	14	46	0.92	2.33	7.67
6	0.8	0	NO	No	20.4	20.4	0.00	14	46	1	2.33	7.67
6	0.7	0	NO	No	23.6	23.6	0.00	14	46	0.92	2.33	7.67
6	0.6	0	NO	No	26.8	26.8	0.00	14	46	1.06	2.33	7.67
6	0.5	0	NO	No	/	30	0.00	60	0	0.63	10	0
6	0.4	0	NO	No	23.99	24	0.04	60	0	0.56	10	0
6	0.3	0	NO	No	/	18	0.00	60	0	0.39	10	0
6	0.2	0	NO	No	/	12	0.00	60	0	0.38	10	0
6	0.1	0	NO	No	6	6	0.00	60	0	0.08	10	0
6	0	0	NO	No	/	/	0.00	60	0	0.06	10	0
									AVG time	0.66		

Figure 4.4: Simulation results for 4-6 vehicles, 12 periods, region of interest with 16 cells (area 4x4), normalised incident density according to the number of vehicles available for assistance

Branch and Bound algorithm used by CPLEX calculates Best Integer solution and Best Bound solution, which are then compared and their mismatch results in parameter

Area 4x4; Tmax=12; Demand normalised according to the No. of Agents

No. of Agents	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
7	1	0	NO	No	28	28	0.00	28	56	1.22	4	8
7	0.9	0	NO	No	30.8	30.8	0.00	28	56	0.8	4	8
7	0.8	0	NO	No	33.6	33.6	0.00	28	56	0.78	4	8
7	0.7	0	NO	No	36.4	36.4	0.00	28	56	0.84	4	8
7	0.6	0	NO	No	39.2	39.2	0.00	28	56	1.08	4	8
7	0.5	0	NO	No	/	42	0.00	79	5	0.52	11.29	0.71
7	0.4	0	NO	No	/	33.6	0.00	84	0	0.44	12	0
7	0.3	0	NO	No	/	25.2	0.00	84	0	0.39	12	0
7	0.2	0	NO	No	/	16.8	0.00	84	0	0.59	12	0
7	0.1	0	NO	No	8.4	8.4	0.00	84	0	0.16	12	0
7	0	0	NO	No	/	/	0.00	84	0	0.06	12	0
									AVG time	0.63		

Area 4x4; Tmax=12; Demand normalised according to the No. of Agents

No. of Agents	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
8	1	0	NO	No	32	32	0.00	32	64	1.42	4	8
8	0.9	0	NO	No	35.2	35.2	0.00	32	64	1.28	4	8
8	0.8	0	NO	No	38.4	38.4	0.00	32	64	1.33	4	8
8	0.7	0	NO	No	41.6	41.6	0.00	32	64	1.03	4	8
8	0.6	0	NO	No	44.8	44.8	0.00	32	64	1.17	4	8
8	0.5	0	NO	No	47.99	48	0.02	92	4	0.97	11.5	0.5
8	0.4	0	NO	No	/	38.4	0.00	96	0	0.59	12	0
8	0.3	0	NO	No	/	28.8	0.00	96	0	0.55	12	0
8	0.2	0	NO	No	/	19.2	0.00	96	0	0.48	12	0
8	0.1	0	NO	No	9.6	9.6	0.00	96	0	0.14	12	0
8	0	0	NO	No	/	/	0.00	96	0	0.06	12	0
									AVG time	0.82		

Area 4x4; Tmax=12; Demand normalised according to the No. of Agents

No. of Agents	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
9	1	0	NO	No	/	36	0.00	36	72	1.64	4	8
9	0.9	0	NO	No	39.6	39.6	0.00	36	72	1.19	4	8
9	0.8	0	NO	No	43.2	43.2	0.00	36	72	1.16	4	8
9	0.7	0	NO	No	46.8	46.8	0.00	36	72	1.27	4	8
9	0.6	0	NO	No	50.4	50.4	0.00	36	72	1.06	4	8
9	0.5	0	NO	No	/	54	0.00	104	4	0.83	11.56	0.44
9	0.4	0	NO	No	/	43.2	0.00	108	0	0.67	12	0
9	0.3	0	NO	No	/	32.4	0.00	108	0	0.67	12	0
9	0.2	0	NO	No	/	21.6	0.00	108	0	0.55	12	0
9	0.1	0	NO	No	10.8	10.8	0.00	108	0	0.22	12	0
9	0	0	NO	No	/	/	0.00	108	0	0.08	12	0
									AVG time	0.85		

Area 4x4; Tmax=12; Demand normalised according to the No. of Agents

No. of Agents	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
10	1	0	NO	No	40	40	0.01	40	80	1.94	4	8
10	0.9	0	NO	No	44	44	0.00	40	80	1.8	4	8
10	0.8	0	NO	No	48	48	0.00	40	80	1.66	4	8
10	0.7	0	NO	No	52	52	0.00	40	80	1.73	4	8
10	0.6	0	NO	No	56	56	0.00	40	80	1.59	4	8
10	0.5	0	NO	No	/	60	0.00	113	7	1.16	11.3	0.7
10	0.4	0	NO	No	/	48	0.00	120	0	0.83	12	0
10	0.3	0	NO	No	/	36	0.00	120	0	0.75	12	0
10	0.2	0	NO	No	/	24	0.00	120	0	0.72	12	0
10	0.1	0	NO	No	12	12	0.00	120	0	0.47	12	0
10	0	0	NO	No	/	/	0.00	120	0	0.09	12	0
									AVG time	1.16		

Figure 4.5: Simulation results for 4-6 vehicles, 12 periods, region of interest with 16 cells (area 4x4), normalised incident density according to the number of vehicles available for assistance

gap being > 0.00%. If CPLEX did not list value of Best Integer or Best Bound, "/" will appear in the results.

As the uncoverage per agent and fatigue per agent generally do not change, without considering the exception, for the number of vehicles from 4 to 10, it can be concluded that all the area cells in the setting 4x4 are covered with minimum 4 vehicles. To test this hypothesis and to be able to make further conclusions, the instances with 1, 2 and 3 vehicles and density normalised accordingly were tested. The results in the Figure 4.6. show the increasing uncoverage per agent with less than 4 vehicles employed, as expected.

Area 4x4; Tmax=12; Demand normalised according to the No. of Agents

No. of Age	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
1	1	0	NO	No	6.59	6.59	0.00	6.59	8	0.08	6.59	8

Area 4x4; Tmax=12; Demand normalised according to the No. of Agents

No. of Age	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
2	1	0	NO	No	8.55	8.76	2.46	8.76	16	0.34	4.38	8

Area 4x4; Tmax=12; Demand normalised according to the No. of Agents

No. of Age	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
3	1	0	NO	No	12.11	12.13	0.14	12.13	24	0.52	4.04	8

Figure 4.6: Simulation results for 1-3 vehicles, 12 periods, region of interest with 16 cells (area 4x4), normalised incident density according to the number of vehicles available for assistance

Simulation analysis with incident density normalised for the $NbVeh = 4$

For further analysis, incident density will be kept constant, and only number of vehicles available to assist will be changed in each instance. The results of instances with the demand kept constant at the incident density normalised for 4 agents available to assist the accident are given in the Figures 4.7 and 4.8, that show the decrease of the uncoverage per agent, as well as the fatigue per agent, as the number of agents increases. It is interesting to see how many vehicles is needed for certain setting in order for all the demand to be covered or supplied, meaning supersaturation of the system is achieved. For the region of interest with 16 cells (area 4x4), 12 periods and incident density normalised for the $NbVeh = 4$ all the incidents are covered with 7 vehicles.

To make some general conclusions about the saturation point, more tests are needed, such as simulating different environment with different area size, different incident den-

Area 4x4; Tmax=12; Demand normalised for 4 agents

No. of Agents	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
5	1	0	NO	No	10	10	0.00	10	39	1.28	2	7.8
5	0.9	0	NO	No	12.8	12.8	0.00	10	38	1.09	2	7.6
5	0.8	0	NO	No	15.6	15.6	0.00	10	38	0.92	2	7.6
5	0.7	0	NO	No	18.4	18.4	0.00	10	38	0.88	2	7.6
5	0.6	0	NO	No	21.18	21.2	0.12	10	38	2.11	2	7.6
5	0.5	0	NO	No	/	24	0.00	41	7	0.7	8.2	1.4
5	0.4	0	NO	No	/	19.2	0.00	48	0	0.42	9.6	0
5	0.3	0	NO	No	/	14.4	0.00	48	0	0.33	9.6	0
5	0.2	0	NO	No	9.6	9.6	0.00	48	0	0.09	9.6	0
5	0.1	0	NO	No	4.8	4.8	0.00	48	0	0.06	9.6	0
5	0	0	NO	No	/	/	0.00	48	0	0.05	9.6	0
									AVG time	0.72		

Area 4x4; Tmax=12; Demand normalised for 4 agents

No. of Agents	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
6	1	0	NO	No	2	4	50.00	4	44	4.11	0.67	7.33
6	0.9	0	NO	No	8	8	0.00	4	44	1.39	0.67	7.33
6	0.8	0	NO	No	12	12	0.00	4	44	1.3	0.67	7.33
6	0.7	0	NO	No	16	16	0.00	4	44	1.48	0.67	7.33
6	0.6	0	NO	No	20	20	0.00	4	44	1.48	0.67	7.33
6	0.5	0	NO	No	/	24	0.00	48	0	0.58	8	0
6	0.4	0	NO	No	/	19.2	0.00	48	0	0.48	8	0
6	0.3	0	NO	No	/	14.4	0.00	48	0	0.5	8	0
6	0.2	0	NO	No	9.6	9.6	0.00	48	0	0.11	8	0
6	0.1	0	NO	No	4.8	4.8	0.00	48	0	0.08	8	0
6	0	0	NO	No	/	/	0.00	48	0	0.05	8	0
									AVG time	1.05		

Area 4x4; Tmax=12; Demand normalised for 4 agents

No. of Agents	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
7	1	0	NO	No	0	0	0.00	0	50	2.09	0	7.14
7	0.9	0	NO	No	4.8	4.8	0.00	0	48	4.06	0	6.86
7	0.8	0	NO	No	9.6	9.6	0.00	0	48	1.97	0	6.86
7	0.7	0	NO	No	14.4	14.4	0.00	0	48	2.19	0	6.86
7	0.6	0	NO	No	19.18	19.2	0.11	0	48	2.31	0	6.86
7	0.5	0	NO	No	/	24	0.00	47	1	0.78	6.71	0.14
7	0.4	0	NO	No	/	19.2	0.00	48	0	0.63	6.86	0
7	0.3	0	NO	No	/	14.4	0.00	48	0	0.55	6.86	0
7	0.2	0	NO	No	9.6	9.6	0.00	48	0	0.13	6.86	0
7	0.1	0	NO	No	4.8	4.8	0.00	48	0	0.11	6.86	0
7	0	0	NO	No	/	/	0.00	48	0	0.05	6.86	0
									AVG time	1.35		

Figure 4.7: Simulation results for 5-7 vehicles, 12 periods, region of interest with 16 cells (area 4x4), incident density normalised for NbVeh=4

sity or different number of time periods. Further analysis will be coducted later in the Chapter 5.

Comparison of the simulation results for normalised incident density and constant incident density for NbVeh=4

Figure 4.9 represents the comparison on the computational complexity level, showing the average computational time between normalised incident density and constant incident density for $NbVeh = 4$. Average computational time is calculated from all the instances with the value of weight ranging from $w = 1$ to $w = 0$, with the change of 0.1. The graph shows that computational time increases in each case proportionally with the increasing number of vehicles. However, it seems that computational complexity is higher for the constant incident density than for the situation when density is normalised according to the number of the available vehicles. In the case for constant incident density it is assumed that there are the supply is higher than demand, which directly

Area 4x4; Tmax=12; Demand normalised for 4 agents

No. of Agents	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
8	1	0	NO	No	0	0	0.00	0	62	0.58	0	7.75
8	0.9	0	NO	No	4.8	4.8	0.00	0	48	1.92	0	6
8	0.8	0	NO	No	9.6	4.8	0.00	0	48	2.38	0	6
8	0.7	0	NO	No	14.4	14.4	0.00	0	48	1.97	0	6
8	0.6	0	NO	No	19.20	19.2	0.00	0	48	2.45	0	6
8	0.5	0	NO	No	/	24	0.00	45	3	0.81	5.63	0.38
8	0.4	0	NO	No	/	19.2	0.00	48	0	1	6	0
8	0.3	0	NO	No	/	14.4	0.00	48	0	0.77	6	0
8	0.2	0	NO	No	9.6	9.6	0.00	48	0	0.14	6	0
8	0.1	0	NO	No	4.8	4.8	0.00	48	0	0.11	6	0
8	0	0	NO	No	/	/	0.00	48	0	0.06	6	0
									AVG time	1.11		

Area 4x4; Tmax=12; Demand normalised for 4 agents

No. of Agents	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
9	1	0	NO	No	0	0	0.00	0	63	0.69	0	7
9	0.9	0	NO	No	4.8	4.8	0.00	0	48	2.66	0	5.33
9	0.8	0	NO	No	9.6	9.6	0.00	0	48	2.92	0	5.33
9	0.7	0	NO	No	14.4	14.4	0.00	0	48	3.34	0	5.33
9	0.6	0	NO	No	19.20	19.20	0.00	0	48	2.34	0	5.33
9	0.5	0	NO	No	/	24	0.00	45	3	1.05	5	0.33
9	0.4	0	NO	No	/	19.2	0.00	48	0	1	5.33	0
9	0.3	0	NO	No	/	14.4	0.00	48	0	0.66	5.33	0
9	0.2	0	NO	No	9.6	9.6	0.00	48	0	0.16	5.33	0
9	0.1	0	NO	No	4.8	4.8	0.00	48	0	0.13	5.33	0
9	0	0	NO	No	/	/	0.00	48	0	0.08	5.33	0
									AVG time	1.37		

Area 4x4; Tmax=12; Demand normalised for 4 agents

No. of Agents	Weight w	Weight y	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	Uncoverage per Ag	Fatigue per Ag
10	1	0	NO	No	0	0	0.00	0	70	1	0	7
10	0.9	0	NO	No	4.8	4.8	0.00	0	48	3.5	0	4.8
10	0.8	0	NO	No	9.6	9.6	0.00	0	48	2.86	0	4.8
10	0.7	0	NO	No	14.4	14.4	0.00	0	48	3.47	0	4.8
10	0.6	0	NO	No	19.20	19.20	0.00	0	48	3.14	0	4.8
10	0.5	0	NO	No	/	24	0.00	44	4	1.59	4.4	0.4
10	0.4	0	NO	No	/	19.2	0.00	48	0	1.11	4.8	0
10	0.3	0	NO	No	/	14.4	0.00	48	0	0.97	4.8	0
10	0.2	0	NO	No	9.6	9.6	0.00	48	0	0.2	4.8	0
10	0.1	0	NO	No	4.8	4.8	0.00	48	0	0.17	4.8	0
10	0	0	NO	No	/	/	0.00	48	0	0.09	4.8	0
									AVG time	1.65		

Figure 4.8: cellults for 8-10 vehicles, 12 periods, region of interest with 16 cells (area 4x4), incident density normalised for NbVeh=4

leads to greater number of coverage possibilities. It seems that CPLEX consumes more time when the possibilities are diversified.

Figure 4.10 is comparing the average uncoverage for different numbers of vehicles. If the uncoverage value for 6 agents is considered as an exception, and having in mind previous analysis, the average uncoverage seems to be increasing linearly with the increase in number of vehicles. On the other hand, the average uncoverage is decreasing with the increase in number of vehicles when the demand is constant. The gradient of this decrease is declining, converging towards zero.

The graph on the Figure 4.11 is similarly showing the average fatigue for different numbers of vehicles. As fatigue part of (3.1) is modeled as a sum of the number of agents, it has an increasing trend for both situations, normalised and constant density. However, the increase with normalised density has a linear trend, while at the constant density the gradient is again declining with the tendency towards zero.

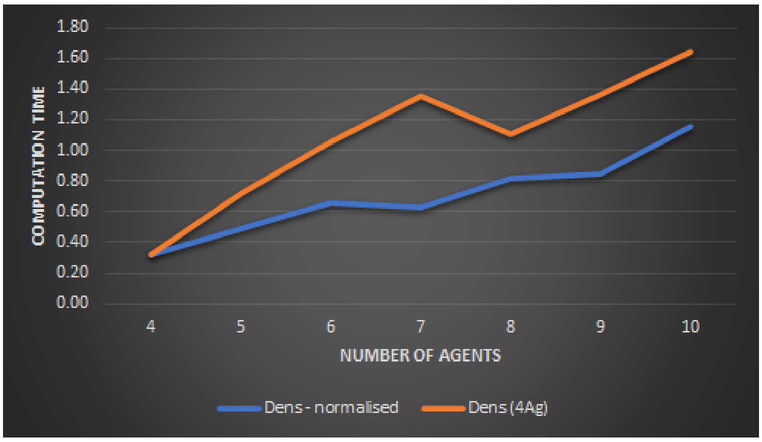

Figure 4.9: Comparison of the average computational time

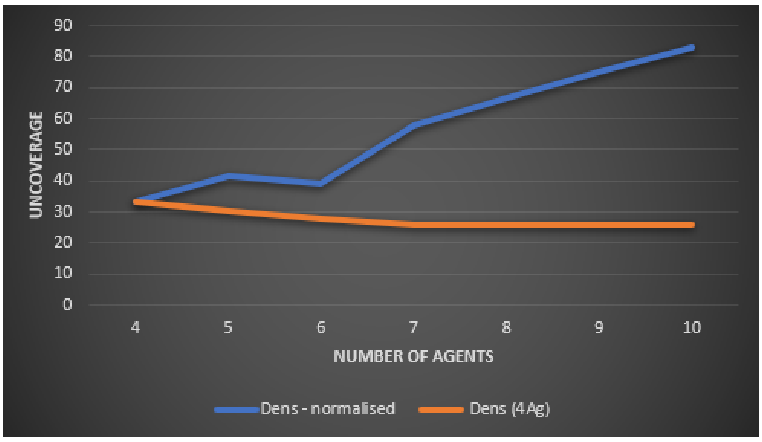

Figure 4.10: Comparison of the average uncoverage

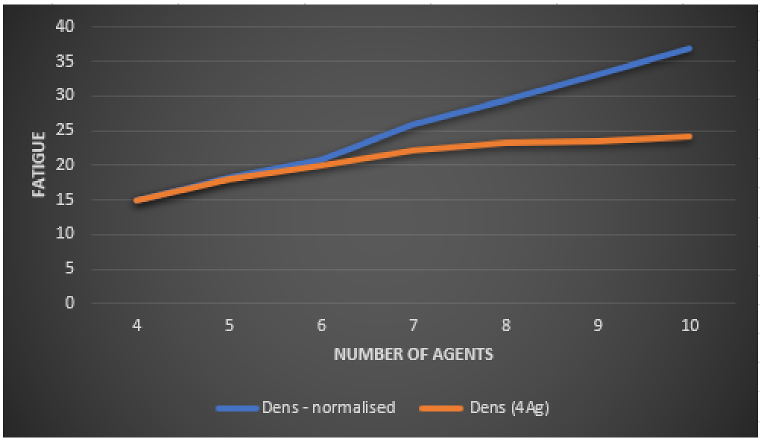

Figure 4.11: Comparison of the average fatigue

5 System balance for given incident density

As previously shown, change in demand, i.e. incident density, with regards to the number of vehicles needed to cover all the incidents, greatly impacts the model. With different demand input, model will result in different output – decision variable values, as well as the change in the model complexity itself. Therefore, it is interesting to investigate how the model output and complexity would change with respect to the set incident density.

"Saturation point" represents the case when supply and demand are balanced, meaning there are enough available vehicles to cover all the estimated demand at the given time, $D_{i\tau} = \sum_{a \in A} z_{a\tau i} \forall i \in \mathcal{I}, \tau \in \mathcal{T}$. "Saturated systems" will have decision variable value of uncoverage $\delta_{i\tau} = 0$.

5.1. System balance for different incident densities

First, incident densities set for 6, 8 and 10 agents have been compared. For this comparison, weight w is held constant with its value being 1. Therefore, only the uncoverage part of the BABPCAC problem is being analyzed. When the demand is changed, the saturation point is interesting for investigation, meaning at which point the system becomes saturated as well as how the saturation affect the model complexity. This is why only area coverage problem needs to be taken into account.

Figure 5.1 represents the change in uncoverage per agent as different number of agents, i.e. vehicles is available to assist the incidents. With increase in number of agents uncoverage/agent drops as expected. Also, when different incident densities are applied, this parameter has higher values and reaches the saturation later. For example, incident density normalised for 6 agents will reach the value of uncoverage/agent being zero when 9 agents are available, while 14 and 17 agents will be needed to cover all the area for incident density being normalised for 8 and 10 agents, respectively. Uncoverage for all densities decreases when more agents are added to the system, finally converges to 0 with the similar trend. Gradient of all three lines is steeper with lower numbers of agents, but is decreasing simultaneously with the uncoverage value.

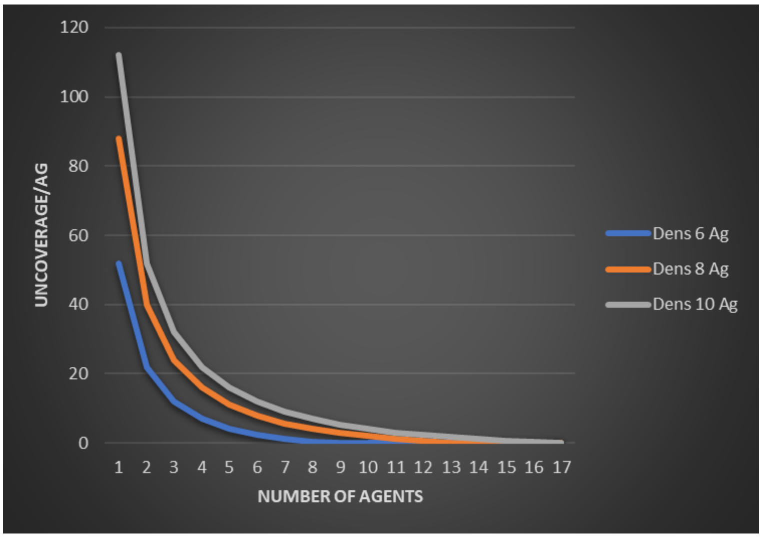

Figure 5.1: Changes in uncoverage per agent for different incident densities

Complexity of the model for the same instances is considered as the next step in the analysis. Model complexity is represented by the computational time needed for solver to propose the optimal solution on the Figure 5.2. There is a clear peak for all instances. This most complex instance for density set for 6 agents is when 8 agents are available, for density set for 8 agents when 12 agents are available, while for density set for 10 agents the most computational time elapsed when 16 agents are available. If referring to Figure 5.1, the most complex instances are the ones with the minimum value of uncoverage, but when uncoverage is still not 0, meaning not yet saturated. Figure 5.2 shows tendency that the most complex instances are near saturation point, while their complexity drops drastically after the saturation point is passed. For example, computational time needed for blue line and 7 agents is 41% of its peak time (for 8 agents) and 15% of its peak time for 9 agents, when system is supersaturated. The graph shows direct relation between the coverage area part of the problem and its complexity.

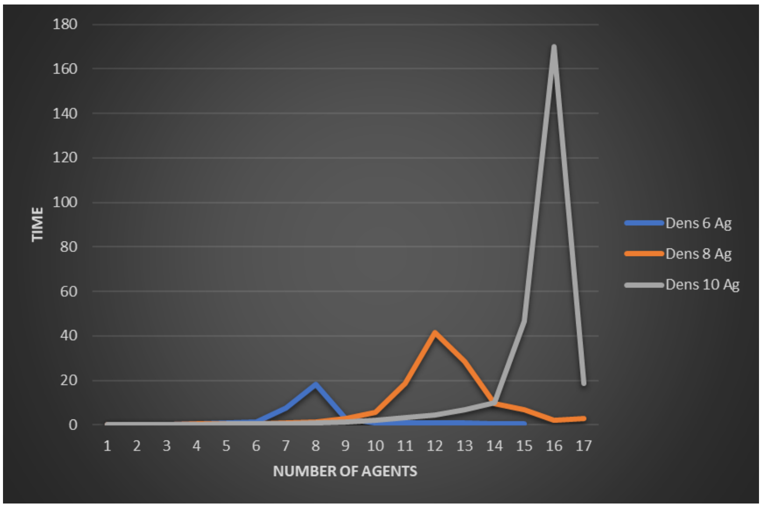

Figure 5.2: Changes in computational time for different incident densities

5.2. System balance for different values of weight w

For further analysis, an instance with demand normalised for 6 agents will be tested. In this step the same instance will be investigated from the perspective of weights applied to the model. If the demand is held constant, and number of vehicles are changed as well as the weight w, what results does the model give. Figure 5.3 shows the change in the variable uncoverage per agent. The model returns the expected results: uncoverage per agent is decreasing exponentially until it reaches certain value where it remains somewhat stable. As the values for $w = 1$ and $w = 0.7$ are the same, Figure 5.4 shows the results when $w = 1$ is applied. As the weight is decreasing, more importance is given to the break scheduling part of the objective function, so is the uncoverage/agent increasing. This is shown very clearly on the both graphs. Uncoverage/agent reaches its peak value for $w = 0.2$ for any given number of vehicles available to assist.

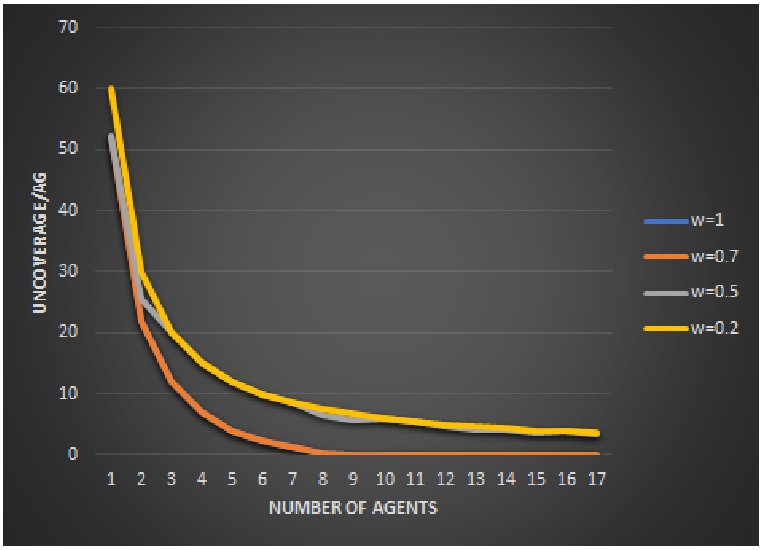

Figure 5.3: Changes in uncoverage per agent for different weight values

Figure 5.5 represents how fatigue per agent is changed with respect to different number of agents, i.e. vehicles as well as different weights applied. Unlike the uncoverage per agent variable, changes and trends are not as smooth here. However, fatigue/agent shows tendency to fall with increasing number of agents, this fall happening earlier and with steeper gradient as the weight associated is reduced; except for when $w = 1$, which is expected as the problem is treated only as area coverage problem, without giving any importance to the fatigue decision variable. Also, for $w = 0.2$, fatigue/agent is remaining 0 regardless of the number of agents available. As previously shown, for all the weight values under 0.4 ($w = 0.4$ included), the model assigns breaks to every agent for every time period, which leads to a sum of fatigue being zero. Weight $w = 0.5$ is critical with its fatigue/agent value drastically dropping from 8 to 0 when only 3 agents are available, without significant changes as the number of agents increases.

Complexity of the model is analyzed through computational time. Figure 5.6 represents the change in computational time needed for CPLEX to solve an optimization problem with different instances as shown above. When value of weight w is less than 1,

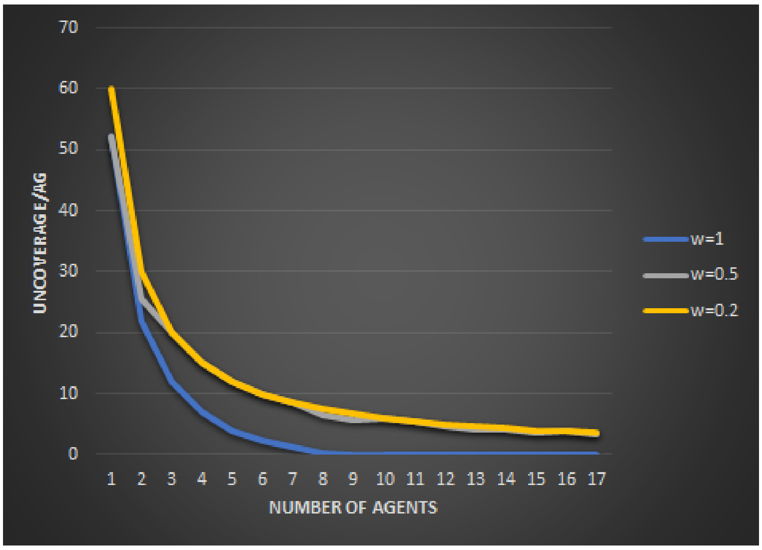

Figure 5.4: Changes in uncoverage per agent for different weight values - emphasis on $w = 1$

that means that model is considering both the problem of uncoverage and fatigue, Figure 5.6 shows rather exponential growth in complexity as the number of vehicles grows. As the fatigue part of the objective function is in fact a sum over all the agents, it is expected that the computational time will grow with increasing number of agents. To show these trends more clearly, data for $w = 0.7$, $w = 0.5$ and $w = 0.2$ is exponentially smoothed and shown on the Figure 5.7 This graph now clearly shows the exponential trends. It is also indicated how this exponential trend is stronger at higher values of w, meaning that the break scheduling part of the problem is consuming less computational time, thus it is less complex to be solved. When applying weight $w = 1$, or when the only focus is on minimizing uncoverage, there is a clear peak on Figure 5.6 when the number of agents employed is 7. As previously shown on the Figure 5.4, this number of vehicles results in minimal uncoverage before the system is supersaturated or fundamentally oversupplied. As soon as this critical point is passed and supply is greater than demand, computational complexity decreases suddenly and stabilizes at lower complexity level. The peak can be explained by how complexity grows with growing number of possible combinations that software needs to run. When supply is relatively low compared to demand, there are not a lot of possibilities that can further optimize the solution, thus the complexity is low as well as the computational time needed to

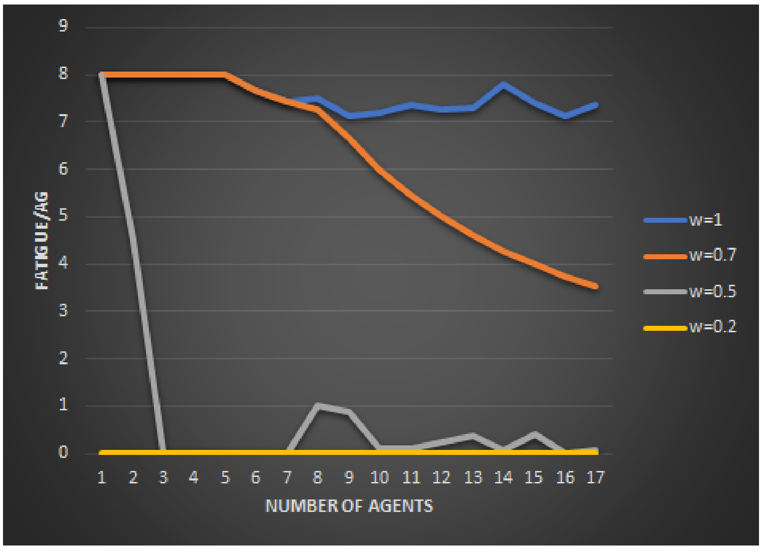

Figure 5.5: Changes in fatigue per agent for different weight values

reach the optimum. The same is happening with supply relatively high compared to demand. However, as soon as the supply is similar, but does not yet exceed the demand, there are numerous combinations that have to be computed and tested in order to get the combination among them that will result in optimal solution. The software is searching for the optimal solution that would give the minimum value, and the more the supply and demand are similar, the more solution combinations are possible but with a little difference between the final solution values among them. After the saturation point is reached, there are also many possible combinations to distribute the resources, but the first found solution is optimal and even redistribution of the resources will not change the optimal value and algorithm stops its search very fast. This point can be seen as the blue line on the Figure 5.6 with the number of agents available for assistance is 9. With 9 agents and weight $w = 1$, model gives a solution with uncoverage being 0. The computational time for this instance is only 15% of the peak computational time, when 8 agents are available. As more agents are being added to the area, complexity is further increasing, but insignificantly. This insight can be helpful when designing the emergency fleet. With knowing a priori the predicted demand for certain time period and particular area, if employing the fleet, as supply, that is slightly greater than given demand, model complexity can be greatly reduced. However, this approach implies that saturation point is known for any given setting. This is why it is needed to further analyze contribution of all three dimensions to the saturation point.

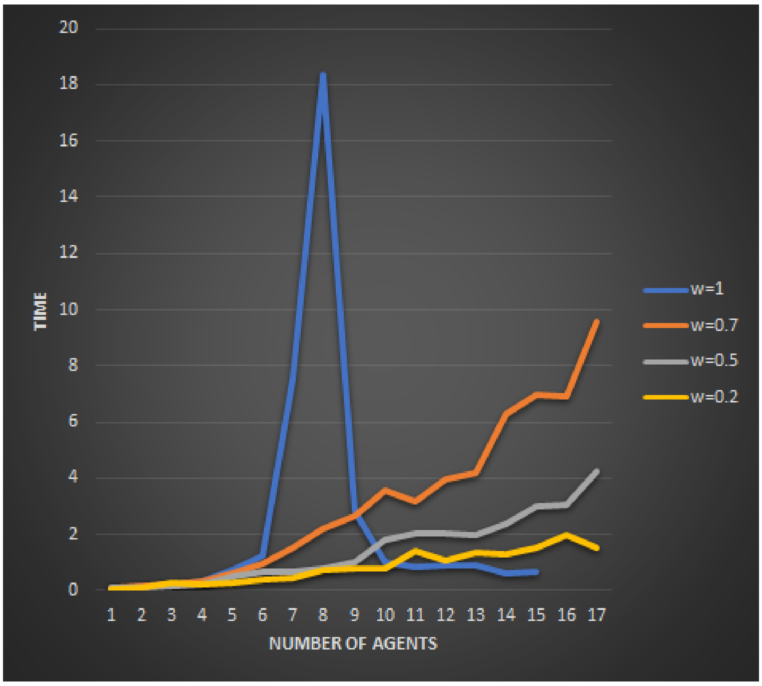

Figure 5.6: Changes in computational time for different weight values

5.3. Trend analysis of the system balance for different dimensions

First analyses have been conducted to investigate trends in saturation when one of the dimensions is being changed, either area size, number of time periods or demand, i.e. incident density. If there are clear trends visible within these dimensions, saturation point could be predicted for any given instance.

All the experiments will be done for $w = 1$, as previously shown that computational complexity decreases in supersaturated systems only if all the importance is given to the uncoverage part of the objective function.

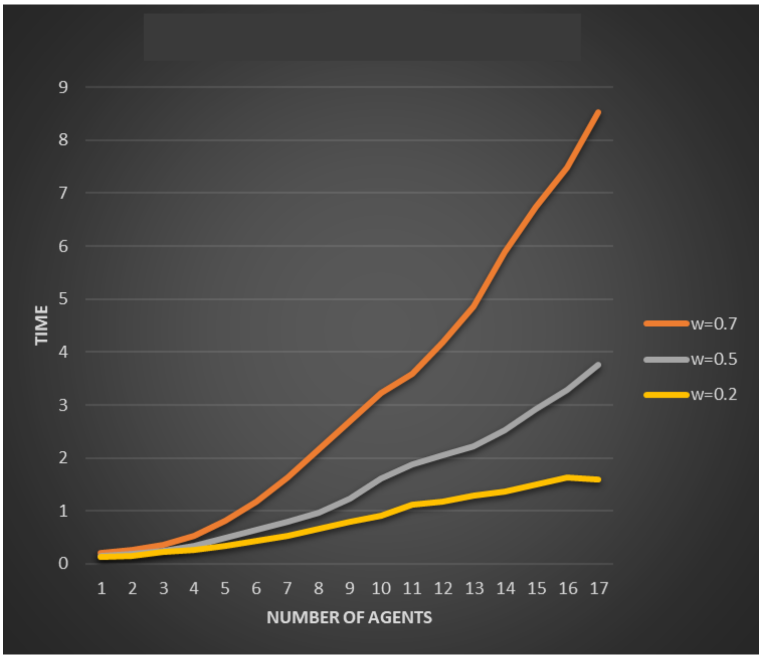

Figure 5.7: Smoothed data for computational time for different weight values

Figure 5.8 represents the change of minimum vehicle number for saturation for two different region of interest size: 16 cells (area 4x4) and 36 cells (area 6x6). If all the other dimensions remain constant, 3 vehicles more are needed for area 6x6 to be completely covered compared to 4x4. That difference remains the same for all the instances, which can lead to conclusion that saturation point is changed proportionally to the area size. Moreover, if the saturation point is reached proportionally to the area dimension, that the very saturation point could be predicted with enough data and further exploration of the effect on the saturation point within the area dimension of the mathematical model.

Change in minimum number of vehicles needed to cover all the demand if different number of time periods is specified as one work shift is shown in Figure 5.9. Only two instances are compared, work shift with 12 and 16 time periods. As there are no clear trends in differences between blue and orange line, no conclusion can be drawn regarding the effect of dimension $Tmax$ on saturation point. Further experiments are needed for more detailed analysis and solid conclusions.

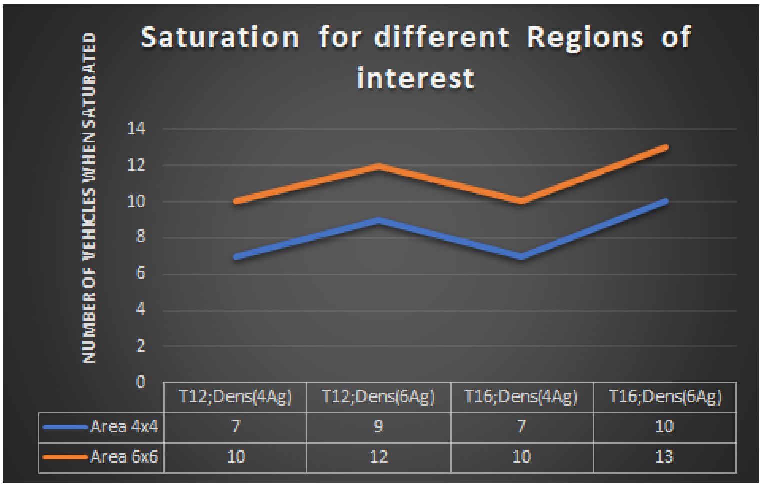

Figure 5.8: Saturation points for different Area dimensions

Comparison of two different incident densities applied as input when all the other dimensions are kept constant is shown in Figure 5.10. Difference between saturation point of estimated demand for 4 and 6 agents is either 2 or 3 vehicles. This comparison slightly clearer than the previous one, with change in the number of time periods, because it shows need for more vehicles employed if estimated demand grows. However, more random instances should be tested in order to draw conclusions about this trend.

Impact of change in all the dimensions should be further explored, but more experiments are needed if statistical trends were to be found. If clear trends exist, a reliable model can be built to help with sizing the fleet of emergency vehicles. That way, optimal number of agents could be employed for each work shift.

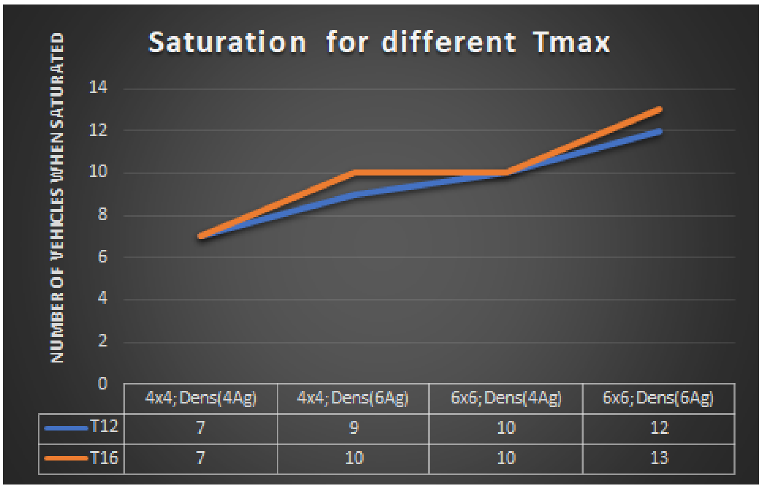

Figure 5.9: Saturation points for different number of time periods $Tmax$

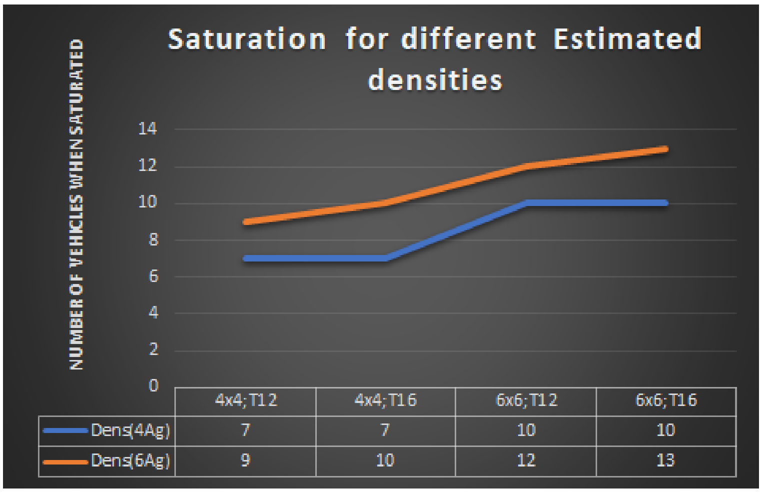

Figure 5.10: Saturation points for different estimated demand (incident density)

6 Scalability of the BAPCAC model

As main application of the BAPCAC model is to define allocation and schedule of emergency fleets, such as ambulance, police, fire service, etc., it is important to ensure low mathematical complexity, so that their schedule is optimised within a reasonable time frame. Therefore, analysis of the model scalability will be conducted. This analysis will lead to discovery of the dimension that impacts computational complexity at the highest level.

There are three dimensions of the BAPCAC model: area size, number of time periods in a work shift and number of agents, i.e. vehicles, that are available for assistance. It is needed to analyze how much each of the three contributes to the mathematical complexity of the model.

Aside from these three main dimensions, there are two other input dimensions that, when changed can differ model output, as well as its complexity: incident density and weight w. Incident density will impact relation between supply and demand, while weight w gives certain importance to either uncoverage or fatigue part of the objective function 3.1.

When sizing the model, if one of the dimensions is changed, number of variables will be changed as well. Thus, it is interesting to see whether a clear and proportional relation between change in model complexity and change in number of variables exists. Correlation analysis will be conducted for this part.

6.1. Correlation analysis

As the first step in correlation analysis, correlation between growing number of variables and constraints will be investigated. As constraints in BAPCAC model are complex and are multidimensional, when one of dimensions is increasing, it is expected that not only number of variables will increase, but also the number of constraints. Therefore, correlation between number of variables and constraints is shown in Figure 6.1 and Figure 6.2. Correlation analysis is conducted separately for region of interest with 16 cells (area 4x4) and region of interest with 36 cells (area 6x6). Both analyses show strong correlation between two parameters, with correlation coefficient $r = 1$ and $r = 0.99$ for area size 4x4 and 6x6 respectively. This proves the initial hypothesis that number of constraints increases proportionally with the increase of number of variables.

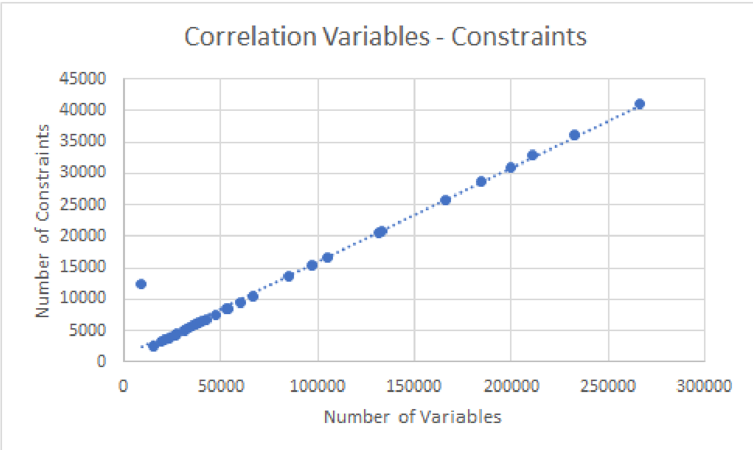

Figure 6.1: Correlation between number of variables and number of constraints; region of interest with 16 cells (area 4x4)

Furthermore, correlation between the number of variables and computational complexity will be investigated. Computational complexity is manifested in computational time or gap. When optimal solution is found within limited time (1 h), gap will be 0 and computational time for CPLEX to give optimal solution for different instances

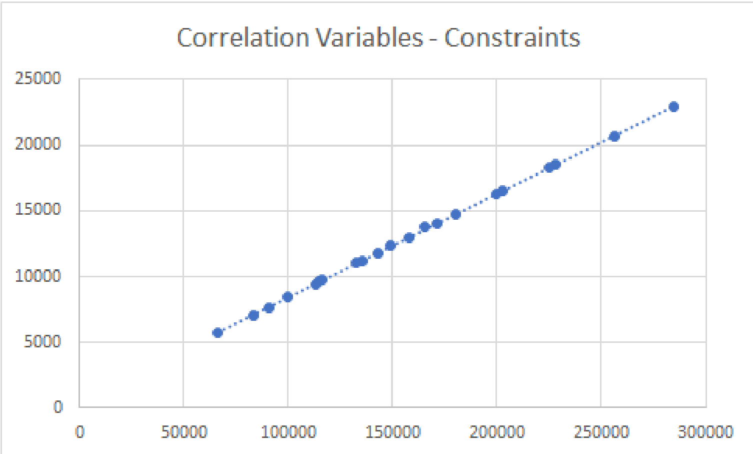

Figure 6.2: Correlation between number of variables and number of constraints; region of interest with 36 cells (area 6x6)

can be compared. On the other hand, if final optimal solution is not found within 1 h, computational complexity will be analyzed through gap between best bound and best integer solution. The more complex certain instance is, the bigger would be the gap.

Therefore, complexity in terms of computational time will be analyzed for instances with region of interest consists of 16 cells (area 4x4), where CPLEX found final optimal solution under 1 h. Figure 6.3 represents correlation between model complexity in terms of computational time and increase in number of variables. Here, correlation coefficient between the two is $r = 0.93$. This strong correlation indicates proportional increase of model complexity with increasing number of variables, i.e. with increase in number of available agents and/or number of time periods.

For instances when region of interest consists of 36 cells (area size 6x6) CPLEX does not propose the optimal solution within 1 h, so only gap for the last feasible solution found can be taken into account when analysing the model complexity. Correlation between gap and number of variables is shown in Figure 6.4. Correlation coefficient is $r = 0.83$, which is slightly lower than correlation for region of interest with 16 cells (area 4x4), but still significant.

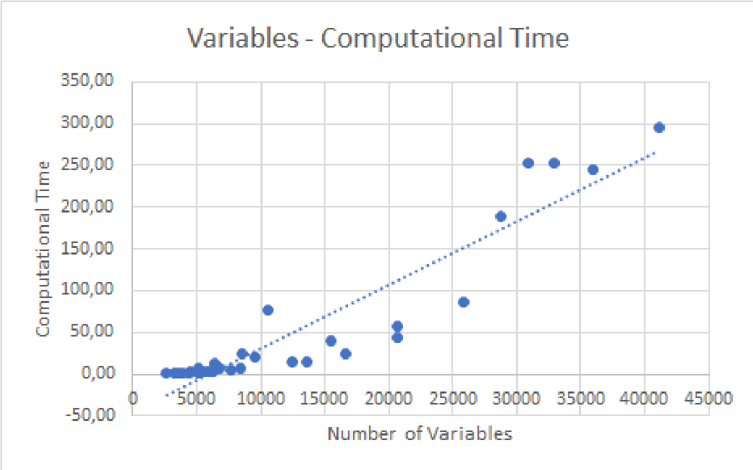

Figure 6.3: Correlation between computational time and number of variables; region of interest with 16 cells (area 4x4)

As the goal was to analyze correlation between change in dimensions and complexity of the model, it is found that number of agents and time periods correlate similarly, whereas change in area size is independent and should be investigated separately. For both analyzes strong correlation between increasing number of variables and model complexity was found. Thus, further scalability analysis will be conducted with number of variables as the reference for model complexity. This is how discrepancy with area size dimension will be avoided.

6.2. Critical weight w

As previously shown, change of weight w also contributes to the model complexity. This is why it is needed to understand its contribution and find critical weight subjective to the BAPCAC model. Critical weight is the one that contributes to the highest computational complexity. Also, it is necessary to understand how the model distributes significance to each of the two parts of the objective function: uncoverage and fatigue.

First, critical weight in terms of model complexity is investigated. Figure 6.5 represents simulation results for two different instances, when number of vehicles is 9 or 10, while demand remains constant. For both cases computational time reaches its maximum when $w = 0.7$. Considering computational time as a reference parameter for BAPCAC model complexity, it seems that values of weight $w = [0.6 - 0.9]$ contribute to

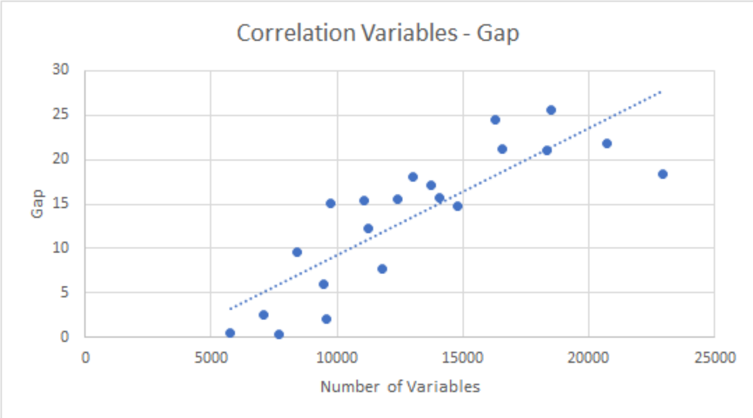

Figure 6.4: Correlation between gap and number of variables; region of interest with 36 cells (area 6x6)

the highest complexity, whereas complexity reaches minimal level when $w = [0 - 0.4]$. These results show higher complexity of the uncoverage part of the function.

Figure 6.5 also shows final values of decision variables, uncoverage and fatigue, when weight is ranging from 0 to 1 with the change of 0.1. For these two instances there is no linear change in decision variables. It seems that all the importance is assigned to uncoverage when weight ranges from 0.6 to 1. There is a slight division of importance at $w = 0.5$ and with weight ranging from 0 to 0.4 all the importance is assigned to fatigue part.

Furthermore, it will be interesting to see how model distributes breaks and allocates vehicles at different values of weight w. Figures 6.6 and 6.7 show that all the agents are assigned with 1 long and 1 short break within duration of the whole work shift, which satisfies all the constraints considering break assignment. Thus, at higher values of weight w BAPCAC model is assigning breaks only for the minimum number of time periods and gives all the importance to the area coverage part of the problem. When $w = 0.5$, break assignment is shown in Figure 6.8. In this case only agent 1 is not on break during the whole work shift, while all the other agents are assigned with breaks for the duration of all 12 periods. Agent 1 is covering time periods with the highest incident density so that importance is distributed in both uncoverage and fatigue. On the other hand, Figure 6.9 is showing break distribution when $w = [0 - 0.4]$, with all

Area 4x4; Tmax=12; Demand normalised for 4 agents								
No. of Agents	Weight w	Weight γ	Deviation	Zbar	Gap [%]	Uncoverage	Fatigue	Real time [sec]
9	1	0	NO	No	0.00	0	63	0.69
9	0.9	0	NO	No	0.00	0	48	2.66
9	0.8	0	NO	No	0.00	0	48	2.92
9	0.7	0	NO	No	0.00	0	48	3.34
9	0.6	0	NO	No	0.00	0	48	2.34
9	0.5	0	NO	No	0.00	45	3	1.05
9	0.4	0	NO	No	0.00	48	0	1
9	0.3	0	NO	No	0.00	48	0	0.66
9	0.2	0	NO	No	0.00	48	0	0.16
9	0.1	0	NO	No	0.00	48	0	0.13
9	0	0	NO	No	0.00	48	0	0.08

Area 4x4; Tmax=12; Demand normalised for 4 agents								
No. of Agents	Weight w	Weight γ	Deviation	Zbar	Gap [%]	Uncoverage	Fatigue	Real time [sec]
10	1	0	NO	No	0.00	0	70	1
10	0.9	0	NO	No	0.00	0	48	3.5
10	0.8	0	NO	No	0.00	0	48	2.86
10	0.7	0	NO	No	0.00	0	48	3.47
10	0.6	0	NO	No	0.00	0	48	3.14
10	0.5	0	NO	No	0.00	44	4	1.59
10	0.4	0	NO	No	0.00	48	0	1.11
10	0.3	0	NO	No	0.00	48	0	0.97
10	0.2	0	NO	No	0.00	48	0	0.2
10	0.1	0	NO	No	0.00	48	0	0.17
10	0	0	NO	No	0.00	48	0	0.09

Figure 6.5: Simulation results for 9 and 10 vehicles; demand normalised for 4 agents

the agents assigned with breaks for all 12 time periods, meaning no incident density is covered. As agents are always on-break for all the lower values of the weight w, there is no need to separately analyze each of these weight values because they will result in the same decision variable values.

Higher values of the weight w will distribute only the minimum number and duration of breaks, so that all the constraints are satisfied and solution is feasible, whereas lower values of weight will automatically assign breaks to vehicles for all the time periods. However, it is more complex to assign particular break periods to agents when uncoverage part has more importance in terms of weight w because BAPCAC model needs to take into account simultaneous minimization of uncoverage. Therefore, it is very important at which time periods agents will work, i.e. cover the incident density, and when they will be assigned 1 short and 1 long break, considering goal for minimizing uncoverage area. This is why the computational time needed for CPLEX to solve the problem is higher for $w = [0.6 - 0.9]$ and $w = 0.7$ will be represented as critical weight with highest computational complexity. Weight $w = 0.7$ will be considered for further analysis of model scalability and dimensional complexity. All the comparisons will be carried out for the critical weight.

Lunch_Breaks... | Value for delta | Value for x | Value for alpha | Value for alpha | Value for delta | Value for y | Value for y

A (size 4)	T (size 12)											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	1	0	0	0	0	1	1	1	0
2	0	0	0	0	1	1	1	0	0	1	0	0
3	0	0	0	0	1	1	1	0	0	1	0	0
4	0	0	0	0	1	0	0	0	1	1	1	0

Figure 6.6: Distribution of breaks for 4 agents and 12 time periods ($w = 1$)

Lunch_Breaks_noZbar_19-10.mod | Area_4x4_Ag4_T12.dat | Value for alpha | Value for delta | Value for x | Value for x | Value for y

A (size 4)	T (size 12)											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	1	1	1	0	0	1	0	0
2	0	0	0	0	1	0	0	0	1	1	1	0
3	0	0	1	0	0	0	0	1	1	1	0	0
4	0	0	0	1	0	0	0	0	1	1	1	0

Figure 6.7: Distribution of breaks for 4 agents and 12 time periods ($w = 0.7$)

Lunch_Breaks_noZbar_19-10.mod | Area_4x4_Ag4_T12.dat | Value for alpha | Value for delta | Value for x | Value for y

A (size 4)	T (size 12)											
	1	2	3	4	5	6	7	8	9	10	11	12
1	1	0	0	0	1	0	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1

Figure 6.8: Distribution of breaks for 4 agents and 12 time periods ($w = 0.5$)

Lunch_Breaks_noZbar_19-10.mod | Area_4x4_Ag4_T12.dat | Value for alpha | Value for delta | Value for x | Value for y

A (size 4)	T (size 12)											
	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1

Figure 6.9: Distribution of breaks for 4 agents and 12 time periods ($w = 0.4 - 0$)

6.3. Critical dimension

Scalability of the BAPCAC model is manifested through the model complexity, either computational time or gap. As previously shown in Section 6.1., model complexity directly correlates with number of variables so model scalability will be analyzed in terms of a critical dimension, that, when increased, contributes to the growing number of variables at the highest level. Also, scalability analysis will be conducted for critical weight $w = 0.7$. Figures 6.10, 6.11 and 6.12 show linear change in number of variables when one of the dimensions (area size, number of time periods $Tmax$, number of vehicles) is increasing while all the other dimensions remain constant.

Three lines in Figure 6.10 show increase in number of variables for three different values of $Tmax$, where each point represents the difference in variable number for area 6x6 compared to 4x4. For example, the lowest point, when $Tmax = 12$ and $NbVeh = 4$ shows that there are 3120 variables more when area is 6x6, compared to the smaller area 4x4 if all the other dimensions are kept constant. If consider the highest linear change in this figure is considered, when $Tmax = 20$ and $NbVeh = 10$, this difference is 12400 between same instance with 4x4 and 6x6. All the simulation results show the increase in number of variables between 3120 and 12400 for all the instances where dimension area is increasing.

Figure 6.11 represents the linear change in number of variables with the increase of $Tmax$ by 4, from $Tmax = 12$ to $Tmax = 16$ and the same values for increase from $Tmax = 16$ to $Tmax = 20$. Each line represents an instance for the constant number of available vehicles. This figure shows that increase in number of variables for increasing number of time periods $Tmax$ ranges from 880 to 4584, which is significantly lower than the increase in dimension area.

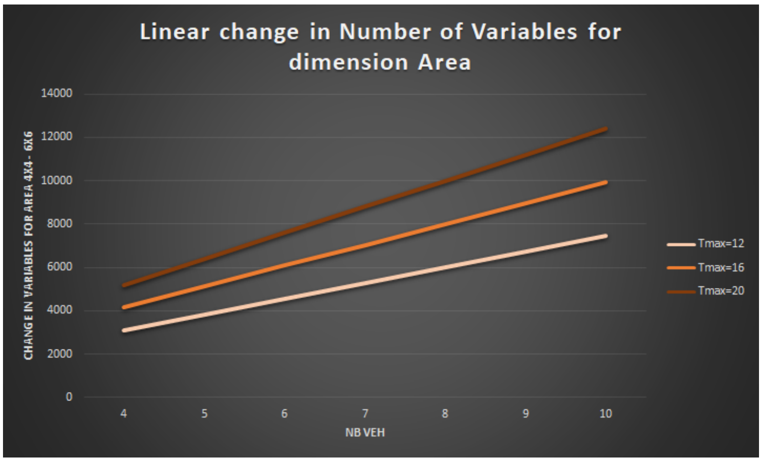

Figure 6.10: Linear change in number of variables for increase in dimension Area

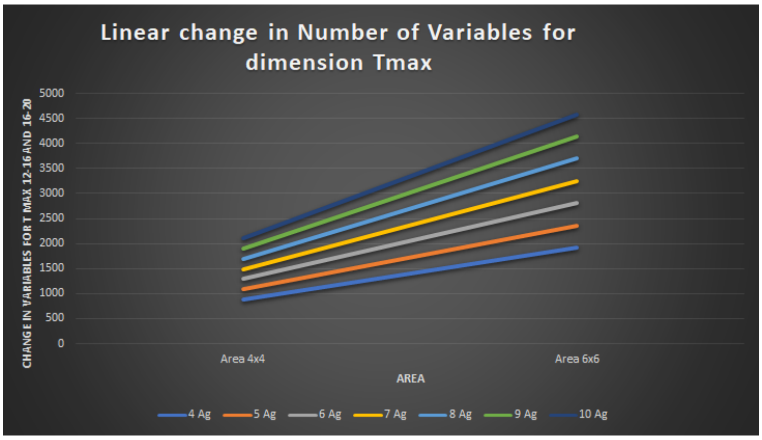

Figure 6.11: Linear change in number of variables for increase in dimension $Tmax$

Contribution of the third dimension, number of vehicles available for assistance, to the model complexity is shown in Figure 6.12, where each of the two green lines represent instances when area size is 4x4 and 6x6. Each point on the graph represents change in number of variables if number of available vehicles increases by 1 (from 1 to 2 vehicles, from 2 to 3 vehicles, etc.). The figure shows that this linear change between different number of vehicles can range from 612 for least complex instances to 2220 added variables when the instance is at the highest complexity level (area 6x6 and $Tmax = 20$). It is clear that dimension with number of vehicles, i.e. agents, contributes to the model complexity less than two other dimensions.

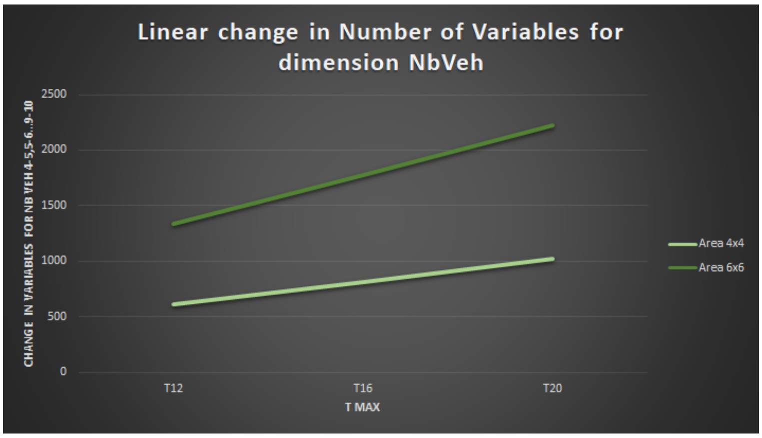

Figure 6.12: Linear change in number of variables for increase in dimension Number of Vehicles

Critical dimension analysis was carried out on a relative basis, comparing experiment instances that have been simulated. Least complex instance was compared to the most complex instance to understand how each of the dimensions contribute to the BAPCAC model complexity as a result of the increase in number of variables.

Regarding scalability, dimensions $Tmax$ and $NbVeh$ are proven to contribute less to the computational complexity than the area size dimension. Therefore, small increase of these two dimensions would not make the model overly complexed for CPLEX, whereas with a minimum increase in area size CPLEX would not be able to find optimal solution within reasonable time, in this case time limited to 1 h.

It is important to emphasize that increase in area size from 4x4 (16 area units) to 6x6 (36 area units) results in adding an extra 20 area units because of its quadratic form. This consequently leads to a larger increase in number of variables. However, this complexity issue should be resolved with a simplification idea.

7 Simplification of the BAPCAC model

To ensure BAPCAC model scalability it is needed to simplify it in terms of the most complex dimension. As analyzed earlier, dimension that contributes the most to computational complexity is area size, precisely instances with area 6x6. Therefore, a simplification that will allow CPLEX to solve instances with bigger areas within reasonable computational time will be proposed. As an example, this simplification will be implemented to instance with region of interest with 36 cells (area 6x6).

7.1. Two-step simplification

In order to simplify BAPCAC model for area dimension, it is necessary to decrease number of variables concerning area. This can be done by simplifying some of the constraints that consist area dimension. However, all of the constraints imposed are essential for proper model setup. Thus, the number of area units needs to be sized down. This is why heuristic approach will be applied with suboptimal solution ensured.

Two-step simplification is suggested as a solution for instance when region of interest has 36 cells (area 6x6). As earlier described, uncoverage part of the objective function 3.1 is more computational complex than fatigue part, especially when area dimension is bigger. This is why proposed solution would solve the problem in two steps. First step will focus only on area coverage in simplified area, while second step will simultaneously solve both area coverage and break assignment problems.

As the first step 36 cells (area 6x6) will be divided into 4 parts, each part consisted of 9 (area 3x3) cells, and will form new area 2x2 with 4 cells, as shown in Figure 7.1. Each color represents one area unit as a sum of 9 area unit with associated incident densities. Within this first step only area coverage part of the problem will be considered and therefore, available agents would be distributed in 4 area units considering only demand associated. When distributed once, agents will not be able to move across area units but will stay in assigned unit for duration of all time periods and will be able to assist only the incidents in area units assigned to them. Regarding constraints, (3.5) and (3.12) need to be adjusted accordingly. Adjacency matrix N_{ij} regulates whether an agent can assist the incident, while adjacency matrix $IM_{jj'}$ limits movement across area units.

The second step will include solving BAPCAC problem for each of the 4 area parts. Each area part, as shown in Figure 7.1, includes 9 area units within area 3x3. Having smaller area size consequently means lower model complexity and thus, less computational time needed. After distributing available agents throughout 4 area parts in the first step, only the smaller number of agents will be considered for each area part in the second step. As second step instances are greatly simplified, optimising model will assign incident to both *idle* and *on-break* vehicles, meaning parameter $\bar{z}_{a\tau i}$ will be included together with $z_{a\tau i}$. This will contribute to minimization of area uncoverage by utilizing small number of agents, possibly employing them even when they are on break.

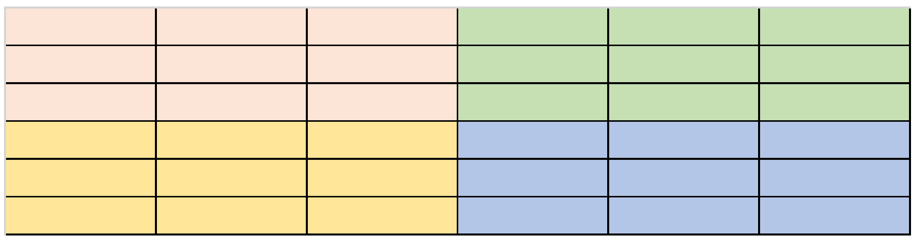

Figure 7.1: Region of interest with 36 cells (area 6x6) divided into 4 cells (area 2x2)

An example of the first step distribution considering area coverage with 5 available agents, i.e. vehicles, is represented in Figure 7.2. In order for coverage to be optimised, CPLEX has distributed all the 5 vehicles across 4 area parts. Because green part has the highest sum of incident densities, 2 vehicles are allocated to that area part, while all the other area parts will have only 1 vehicle at disposal for the second step of the optimization problem.

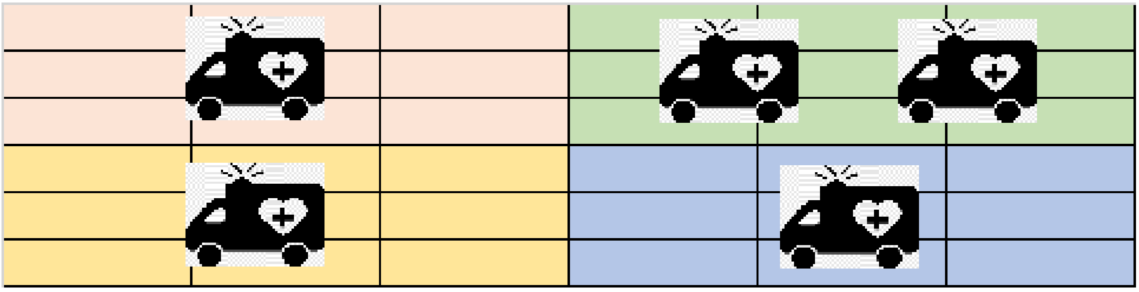

Figure 7.2: Distribution of 5 agents in 4 area parts

7.2. Simulation results for the simplified model

To test performance of the simplified model, its simulation results will be compared to the initial model results. Comparison will be performed with focus on model complexity as well as model output, meaning decision variables uncoverage and fatigue.

Instance with 5 agents available, 16 time period and region of interest with 36 cells (area 6x6) will be considered. First, simulation results for model with no simplification are shown in Figure 7.3. When more importance is given to the uncoverage part of the problem, CPLEX does not find an optimal solution within limited time of 1 h. Focus is therefore put on resolving this particular problem.

No. of Agents	Weight w	Weight gamma	Deviation	Zbar	Best Bound	Best Integer	Gap [%]	Uncoverage	Fatigue	Real time [sec]	No. of Agents	Weight w
5	1	0	NO	No	11,1878	13,2668	15,67	13,267	53	3600,06	5	1
5	0,7	0	NO	No	23,7524	25,2733	6,02	13,39	53	3600,11	5	0,7
5	0,5	0	NO	No	/	32	0,00	64	0	15,33	5	0,5
5	0,2	0	NO	No	12,8	12,8	0,00	64	0	0,81	5	0,2

Figure 7.3: Simulation results with no simplification (16 time periods, 5 agents, region of interest with 36 cells (area 6x6))

First step simplification results, area coverage for 4 area parts, is shown in Figure 7.4. Weight is set to be $w = 1$ because only uncoverage should be considered here. Figure 7.5 shows distribution of 5 vehicles in simplified area. The same distribution will be considered for the second step of the simplified model. Initial uncoverage is lower compared to the results of model without simplification and computational complexity is minimal with computational time needed for optimal solution being only 0.06 sec.

Weight w	Uncoverage	Best Bound	Best Integer	Gap [%]	Uncoverage	Real time [sec]
1		/	4,7529	0,00	4,7529	0,06

Figure 7.4: Simulation results for the first step simplification - area coverage (16 time periods, 5 agents, region of interest with 4 cells (area 2x2))

1 veh	2 veh
1 veh	1 veh

Figure 7.5: First step simplification distribution results (16 time periods, 5 agents, region of interest with 4 cells (area 2x2))

As the second step involves 4 separate simulations needed, sum of their solutions will be taken into account while analyzing the effectiveness of proposed model simplification. Results of the second step are represented in Figure 7.6. Sum of both computational complexity and decision variable solutions is improved if two-step simplification is applied.

No. of Agents	Weight w	Weight gamma	Deviation	Zbar	Gap [%]	Uncoverage	Fatigue	Real time [sec]	No. of Agents	Weight w
5	1	0	NO	Yes	9,45	7,0097	48	2,01	5	1
5	0,7	0	NO	Yes	5,11	15,6951	1	0,97	5	0,7
5	0,5	0	NO	Yes	5,83	16,1848	0	0,53	5	0,5
5	0,2	0	NO	Yes	1,90	16,1848	0	0,4	5	0,2

Figure 7.6: Second step simplification results as sum of the 4 area part simulations

Comparison of decision variable results is provided in Figures 7.7 and 7.8. Uncoverage is lower for all weights but critical $w = 0.7$, which has negligibly higher uncoverage for simplified model. Regarding fatigue, it is considerably lower for higher values of weight w if simplification is implemented. This is a result of assigning incidents to *on-break* vehicles as well, i.e. including parameter $\bar{z}_{a\tau i}$.

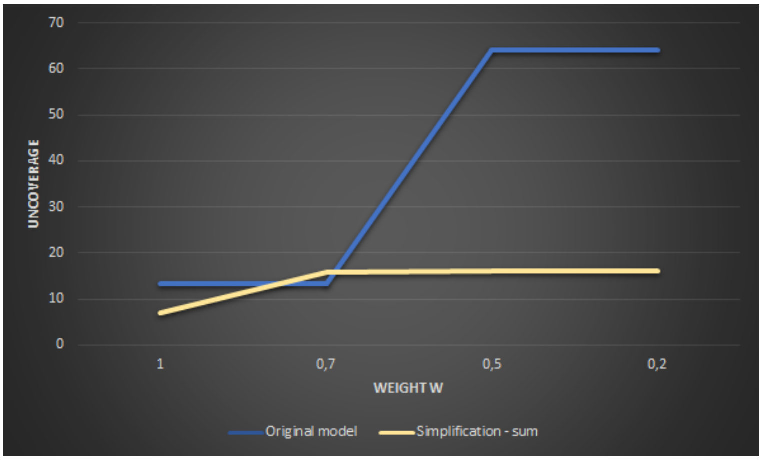

Figure 7.7: Comparison of uncoverage with and without model simplification

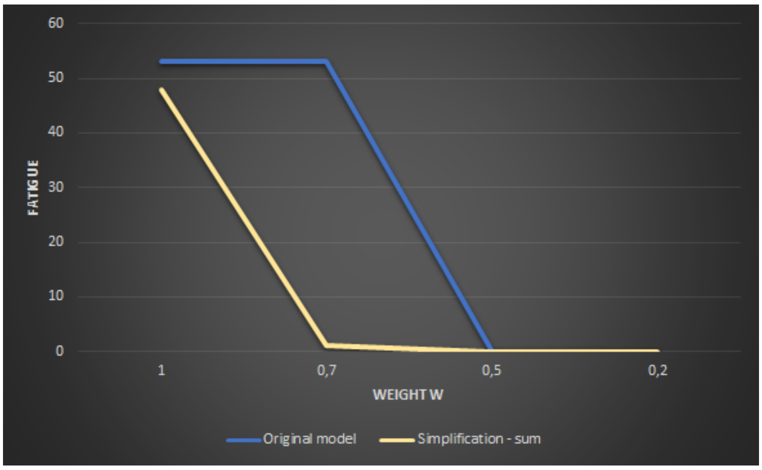

Figure 7.8: Comparison of fatigue with and without model simplification

Computational complexity is compared in Figures 7.9 and 7.10, comparing gap and computational time respectively. Gap is significantly higher at $w = 1$ and $w = 0.7$ for model without simplification. Even though gap values are lower for $w = 0.5$ and $w = 0.2$ for initial model, simplified model results are given as summation of acceptably low gap values. Also, when comparing computational time simplified model needs remarkably less time. Zoomed representation of computational time for lower values of weights w is shown in Figure 7.11 If simplified model is implemented, sum of all 4 area part simulations, time does not exceed 2 seconds. Even with small gap value results, simplified model represents significant reduction in model complexity, which was the main goal of this work.

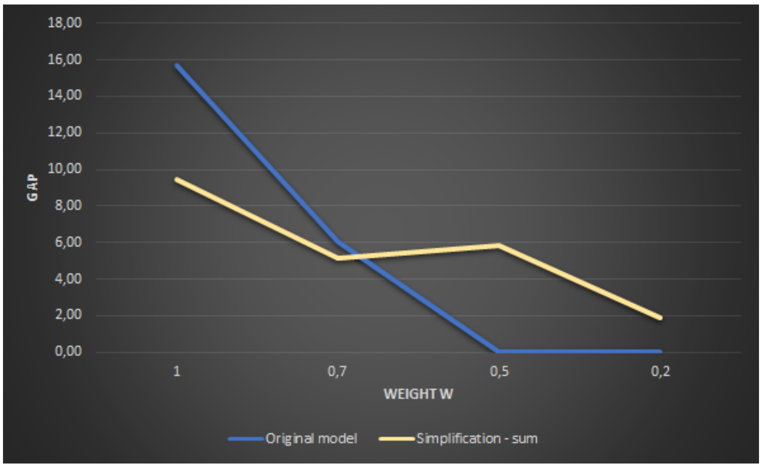

Figure 7.9: Comparison of gap with and without model simplification

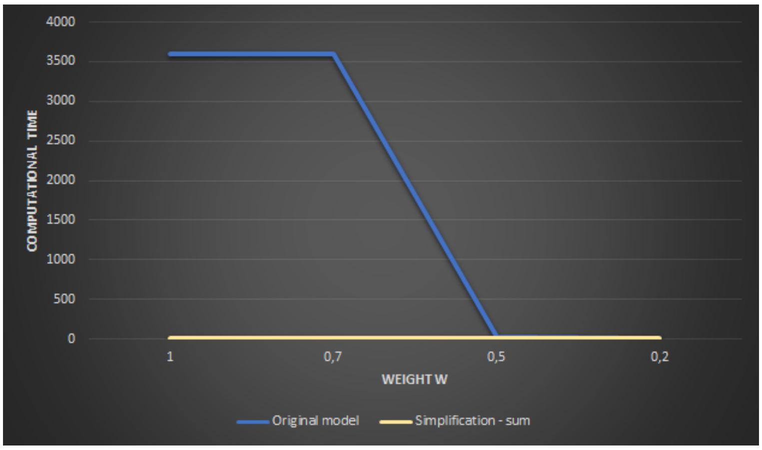

Figure 7.10: Comparison of computational time with and without model simplification

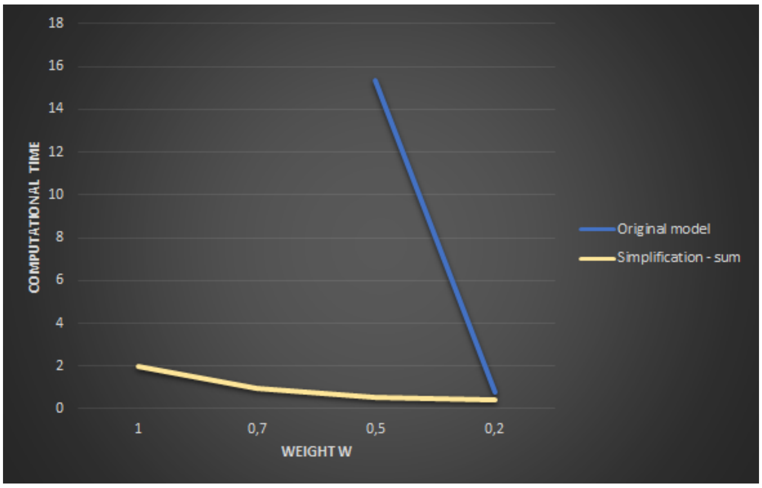

Figure 7.11: Zoomed comparison of computational time with and without model simplification for lower values of weight w

8 Conclusion

This thesis investigated computational limits of mathematically modeled break assignment problem (BAPCAC) considering area coverage, which addresses both break scheduling problem (BSP) and maximal coverage location problem (MCLP) at once.

The main purpose of BAPCAC problem is its implementation in systems with high urgency for coordination of a fleet with regards to the occuring demand. This imposes requirement for optimal solution of BAPCAC model to be found within 1 h. To ensure the reasonable level of computational complexity, scalability of the model, as well as its output have been analyzed.

All the conclusions were based on simulation experiment results using the optimization solver IBM ILOG CPLEX Optimization Studio. Results have been compared on a relative basis since it is important to take into account speed of the processor in a computer used for simulation experiments.

Three dimensions were evaluated: area size, fleet size and time horizon size. Among these, it is found that area size dimension contributes to the computational time, thus model complexity, on the highest level.

Two-step BAPCAC model simplification, focused on reducing computational cost of the area size dimension, is proposed and evaluated. This simplification resulted in a significant reduction of the model complexity, offering the possibility of solving the problem almost in real time. Decision variable values for area uncoverage and agents fatigue were also decreased in comparison to the initial model results.

Future analysis of the BAPCAC model should consider various system settings. This can involve different spatial arrangements of the area, improved break parameters considering minimal engagement of each agent etc. Depending on particular needs of the system that is being modeled, BAPCAC model is sufficiently flexible and able to respond to all the newly imposed requirements, while not contributing significantly to the computational cost.

Bibliography

[1] M. Lujak; A.G. Sànchez; M.O. Mier; H. Billhardt. *An Overview of a Break Assignment Problem Considering Area Coverage*. https://hal.archives-ouvertes.fr/hal-03104345, 2021.

[2] M. Lujak and B. Billhardt. *Case Study from Short Term Scientific Mission "Emergency fleet break assignment problem considering area coverage"*, page 30. Handbook "What is Industrial Mathematics? Case Studies from MI-NET. Mathematics for Industry Network (MI-NET)"COST ACTION TD1409., 2019.

[3] M. Lujak. *STSM "Break assignment in emergency fleets"*, page 41. Program Abstracts Book. ICIAM 2019 Valencia, 9th International Congress on Industrial and Applied mathematics, July 15-19, Valencia, Spain, 2019.

[4] M. Widl and N. Musliu. *The break scheduling problem: complexity results and practical algorithms*, pages 97—112. Memetic Comp. 6, 2014.

[5] R. Church and C. ReVelle. *The Maximal Covering Location Problem.* Addison-WesleyPapers of the Regional Science Association, 1974.

[6] T.R. Robbins and T.P. Harrison. *A stochastic programming model for scheduling call centers with global Service Level Agreements*, pages 1608–1619. European Journal of Operational Research 207, 2010.

[7] H. Fei; N. Meskens and C. Chu. *A planning and scheduling problem for an operating theatre using an open scheduling strategy*, pages 221–230. Computers & Industrial Engineering 58, 2010.

[8] U. Aickelina and K.A. Dowslandb. *An indirect Genetic Algorithm for a nurse-scheduling problem*, pages 761–778. Computers & Operations Research 31, 2004.

[9] L. Di Gaspero; J. Gärtner; N. Musliu; A. Schaerf; W. Schafhauser and W. Slany. *Automated Shift Design and Break Scheduling*, volume 505 of *Automated Scheduling and Planning. Studies in Computational Intelligence*. Springer, Berlin, Heidelberg, 2013.

[10] M. Reikk; J.-F. Cordeau and F. Soumis. *Implicit shift scheduling with multiple breaks and work stretch duration restrictions*. Springer Science+Business Media, LLC, 2009.

[11] C.G. Quimper and L.M. Rousseau. *A large neighbourhood search approach to the multi-activity shift scheduling problem*, pages 373—392. J Heuristics 16, 2010.

[12] A. Gallais and J. Carle. *An Adaptive Localized Algorithm for Multiple Sensor Area Coverage*, pages 525–532. 21st International Conference on Advanced Information Networking and Applications (AINA '07), Niagara Falls, ON, 2007.

[13] L. Brotcorne; G. Laporte and F. Semet. *Ambulance location and relocation models*, pages 451–463. European Journal of Operational Research 147, 2003.

[14] M. Gendreau; G. Laporte and F. Semet. *A dynamic model and parallel tabu search heuristic for real-time ambulance relocation*, pages 1641–1653. Parallel Computing 27, 2001.

[15] M. Gendreau; G. Laporte and F. Semet. *The maximal expected coverage relocation problem for emergency vehicles*, pages 22–28. Journal of the Operational Research Society 57, 2006.

[16] M.S. Maxwell; M.Restrepo; S.G. Henderson and H. Topaloglu. *Approximate dynamic programming for ambulance redeployment*, pages 266–281. INFORMS Journal on Computing 22, 2010.

[17] J. Naoum-Sawaya and S. Elhedhli. *A stochastic optimization model for realtime ambulance redeployment*, pages 1972–1978. Computers Operations Research 40, 2013.

[18] B. Alrifaee; M.G. Mamaghani and D. Abel. *Centralized Non-Convex Model Predictive Control for Cooperative Collision Avoidance of Networked Vehicles.* 2014 IEEE International Symposium on Intelligent Control (ISIC), 2014.

[19] J.D. Ullman. *NP-Complete Scheduling Problems.* Journal of Computer and System Sciences 10, 384–393, 1975.

[20] S. Burer and A.N. Letchford. *Non-convex mixed-integer nonlinear programming: A survey.* Surveys in Operations Research and Management Science, 2012.

[21] IBM ILOG CPLEX Optimization Studio. *https://www.ibm.com/products/ilog-cplex-optimization-studio.* [cited 2020 Dec 28].