Control & Guidance of Multiple Air-Vehicle Systems (MAS)

Research Program POD0914267

Performed for

DRTech, Ministry of Defence

by:

Temasek Laboratories @NUS

with

Department of Electrical & Computer Engineering, NUS

School of Electrical & Electronics Engineering, NTU

Technical Report No. TL/CG/2014/04

Progress Report for Deliverable D8

March 28, 2014

Part IV

Task Assignment Problem for UAVs.

S.M. Dymkou

Aeronautical Sciences Division, Control Science Group,

Temasek Laboratories, National University of Singapore,

T-Lab Building 5A, Engineering Drive 1,

Singapore 117411

Contents

Chapter 1

Introduction

1.0.1 Specific problem formulation

General description : To assign air-vehicles to perform as many simultaneous service requests as possible.

Starting conditions :

Given :

1. The MAS is performing coverage
2. Miltipble requests for service with the following inormation:
   - Number of air-vehicles required
   - Location where air vehicles need to visit
   - Earliest time of 1-st visit
   - Latest time of 1-st visit
   - Minimum duration per visit
   - Maximum interval between visits
   - Time of last visit

Mission Objective :

Find :

1. Air-vehicle(s) to be assigned to request and the corresponding paths to take to the service request
2. Variations to requests with minimal change if the request cannot be met

That :

1. Maximises the number of service requests that can be serviced

Constraints :

Subject to :

1. Air-vehicle performance and dynamics
2. Sensor performance
3. Air to Air datalink performance
4. LOS occlusion in area of operations
5. At least one air-vehicle being directly connected to GCS 90 procents of the time (for MAS to be provided feedback to GCS)

From mathematical point of view this problem can be reformulated as extremal problem which is includes nonlinear systems, PDE and ODE, conflicting situation, vector cost functions, incomplete information, constraints. It is clear that effective algorithms for the solution of this problem must be effective if we use them to solve:

- optimization problems with one costs function;
- optimization problem for ODEs;
- optimization problem for linear ODEs;
- optimization problems for linear discrete processes;
  Thus the logic in consideration inevitable leads to
- linear programming.

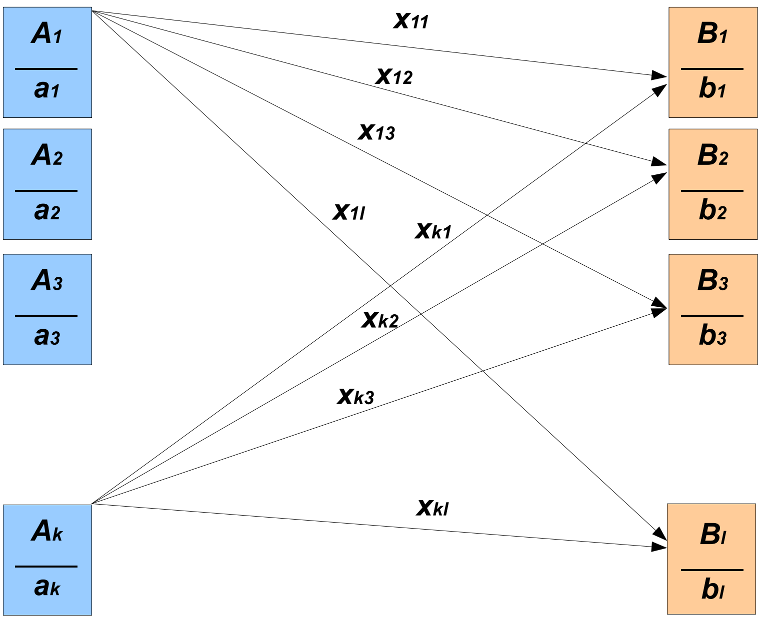
By this reason for the formulated problem above in a reduced form the special class of linear programming (LP) method will be considered in Chapter 2 for static cases. Then the new supporting optimization algorithm (SOA) which is based on constructive approach for using an Integer Linear Program (ILP) formulation to find the optimal solution to multiple-task assignment problem where the tasks are coupled by timing and task order constraints will be presented in the Chapter 3-5 for dynamical cases, where multiple unmanned air vehicles (UAVs) are required to prosecute geographically dispersed target (zones). The UAVs must perform the task on each zone. The optimal performance of these tasks requires cooperation amongst the UAVs such that the critical constraints are satisfied. In chapter 6, we are consider the alternative approach, so called dynamical programming apprach for the same class of problems. And finally in a Chapter 7 and 8 we will introduce the decentralized algorithms: Contract Net Protocols and Consensus Based Bundle method, for the cases, when all previous approaches are not useful by some reasons (i.e. communication problems between UAVs-GCS, UAV-UAV and etc).

Chapter 2

LP assignment problem for group of UAVs

2.1 Formal problem statement

Assume that we have k aerobases $A_1, A_2, ..., A_i, ..., A_k$. Denote by $a_1, a_2, ..., a_i, ..., a_k$ their capacity, namely the maximal number of homogenous UAVs located in aerobase. Also we have l zones of area of operation $B_1, B_2, ..., B_j, ..., B_l$. It is assumed that each onetime service of each zone B_j requests includes at least b_j numbers of UAVs, $j = 1, \ldots, l$.



Also assume that the sum of all requests are equal to the total number of available UAVs.

$$\sum_{i=1}^{k} a_i = \sum_{j=1}^{l} b_j \tag{2.1}$$

Next, for all i and j denote by c_{ij} the benefit of sending the UAV from i-th aerobase to j-th zone of area of operation. Note that the benefit of service can be given by different values, for example, by service time T_{ij}, fuel consumption, total number of UAVs involved etc. Then the problem is to define the plan of service for UAVs, in order to complete all incoming requests for service with maximal benefits. Denote by x_{ij} the number of UAVs from i-th aerobase which are send to service j-th zone. Then using our notation we can formulate the problem statement as the following integer programming problem: To find $x_{ij}, (i = 1, 2, ..., k; j = 1, 2, ..., l)$ such that, the total cost function for all services performed by all UAVs takes a maximal value

$$F = \sum_{i=1}^{k} \sum_{j=1}^{l} c_{ij} x_{ij} \to \min_{x_{ij}} \tag{2.2}$$

subject to

$$\begin{aligned} \sum_{i=1}^{k} x_{ij} = b_j, \quad j = 1, 2, ..., l \\ \sum_{j=1}^{l} x_{ij} = a_i, \quad i = 1, 2, ..., k \\ \sum_{i=1}^{k} a_i = \sum_{j=1}^{l} b_j \\ x_{ij} \geq 0, \qquad x_{ij} \ \text{ are integer numbers.} \end{aligned} \tag{2.3}$$

Here the first two constraints means that we are need to satisfy all service request from each area of operation and to use for that purpose all available UAVs.

Remark 1.

Also it should be noted that in case of inequalities constraints instead of 8.2, the problem can be reduced to the equality case by introducing an additional variables. For example, for $\sum\limits_{i=1}^{k} a_i \geq \sum\limits_{j=1}^{l} b_j$ we will need to introduce some "artificial" aerobase or area of operation with correspondent capacity(for aerobase) and requirements (for area of operation):

$$b_{l+1} = \sum_{i=1}^{k} a_i - \sum_{j=1}^{l} b_j$$
$$a_{k+1} = \sum_{j=1}^{l} b_j - \sum_{i=1}^{k} a_i$$

According to the given Remark 1 it is assumed below that the constraints of (8.2) hold as equalities.

Remark 2. There are a lot of methods to solve such kind of problem. In this report the formulated simple integer optimization problem will be extended to search service schedular for dynamical case of service requests. For this purpose a new supporting optimization method will be developed on the base of the so-called constructive approach by Gabasov R. and Kirillova F.M. [2]. For linear programming problem this method was compared with classical simplex method in the paper [**?**], also was applied in gas industry [5] The proposed method allows to use both initial information and current one produced during the solution process. This together with developed for this case duality theory leads to high efficiency of the associated numerical algorithms. Here we will present a specific case of simplex method developed for assignment problem and transportation problems. This method include the following basic steps:
1. To find initial plan,
2. Check optimality condition for that plan,
3. Construct the improved plan in case of nonoptimality.
In order to demonstrate a key elements of this method, let us to consider the following example:

Example: Assume that three, five, and two homogeneous UAVs located on three aerobases correspondently. It is necessary to send them into 4 area of operations (zones), namely four UAVs into zone number 1, two UAVs into zone number 2, three UAVs into zone number 3 and one UAV into zone number 4. The distances between aerobases and zones are known, and given in the following table:

	B_1	B_2	B_3	B_4
A_1	400	600	800	200
A_2	400	1200	500	100
A_3	800	1000	600	400

Table 1: Distances between aerobases A_i and area of operations B_j

The problem is to define the plan of service of area of operation by UAVs, in order to complete all incoming requests for service with minimal fuel consumption. (Average fuel consumption for distance unit is 0,01 units of fuel.)

These problem can be formulated as follows:

$$F = \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} x_{ij} \to \min_{x_{ij}} \tag{2.4}$$

subject to

$$\begin{gathered} \sum_{i=1}^{3} x_{ij} = b_j, \quad j = 1,2,3,4 \\ \sum_{j=1}^{4} x_{ij} = a_i, \quad i = 1,2,3 \\ \sum_{i=1}^{3} a_i = \sum_{j=1}^{4} b_j \\ x_{ij} \geq 0, \quad x_{ij} \in \mathbb{N}. \end{gathered} \tag{2.5}$$

The condition of that problem can be represented in table form:

	B_1	B_2	B_3	B_4	a_i
A_1	x_{11}	x_{12}	x_{13}	x_{14}	$a_1 = 3$
A_2	x_{21}	x_{22}	x_{23}	x_{24}	$a_2 = 5$
A_3	x_{31}	x_{32}	x_{33}	x_{34}	$a_3 = 2$
b_j	$b_1 = 4$	$b_2 = 2$	$b_3 = 3$	$b_4 = 1$	$\sum_{i=1}^{3} a_i = \sum_{j=1}^{4} b_j = 10$

Below we give the detailed step-by-step procedure to determine the optimal solution.

2.2 Initial feasible solution

To construct the initial feasible solution we will use "North-West corner" method. The construction of the initial supporting feasible solution consist from the several steps on each of them are filled either a row or a table column. The procedure begins with the left top ("northwest") element x_{11} of the plan.

$$x_{11} = \min(a_1; b_1) \tag{2.6}$$

If $a_1 < b_1$, i.e. $x_{11} = a_1$, than from the further consideration we eliminate all elements from the first row. If $a_1 \geq b_1$, i.e. $x_{11} = b_1$, than all elements from the first column are eliminated. In the case $a_1 < b_1$ the next element of feasible solution will be chosen from the second row by the rule $x_{21} = \min(a_2; b_1 - a_1)$. Next, if $a_2 < b_1$, i.e. $x_{21} = a_2$, and in this case we eliminated from our further consideration all elements from the second row. If $a_2 \geq b_1 - a_1$, i.e. $x_{21} = b_1 - a_1$, and further we will not consider the elements from the first column. The next assignment will be made on the intersection of the second column and second row as follows: $x_{22} = \min(a_1 + a_2 - b_1; b_2)$. Then repeated this procedure we will find all elements of the initial supporting feasible solution.
In our case we have the following:

$$\begin{aligned}
x_{11} &= \min(a_1; b_1) = \min(3; 4) = 3 \\
\Downarrow & \quad \text{(next consider the element from the second row)} \\
x_{21} &= \min(a_2; b_1 - a_1) = \min(5; 1) = 1 \\
\Downarrow & \quad \text{(consider second column, element } A_2B_2 \text{)} \\
x_{22} &= \min(a_1 + a_2 - b_1; b_2) = \min(4; 2) = 2 \\
\Downarrow & \quad \text{(consider third column, element } A_2B_3 \text{)} \\
x_{23} &= \min(a_1 + a_2 - b_1 - b_2; b_3) = \min(2; 3) = 2 \\
\Downarrow & \quad \text{(consider third row, element } A_3B_3 \text{)} \\
x_{33} &= \min(a_3; b_3 - (a_1 + a_2 - b_1 - b_2)) = \min(2; 1) = 1 \\
\Downarrow & \quad \text{(consider fourth column, element } A_3B_4 \text{)} \\
x_{34} &= \min(a_1 + a_2 + a_3 - b_1 - b_2 - b_3; b_4) = \min(1; 1) = 1
\end{aligned} \tag{2.7}$$

	B_1	B_2	B_3	B_4	a_i
A_1	$x_{11} = 3$				$a_1 = 3$
A_2	$x_{21} = 1$	$x_{22} = 2$	$x_{23} = 2$		$a_2 = 5$
A_3			$x_{33} = 1$	$x_{34} = 1$	$a_3 = 2$
b_j	$b_1 = 4$	$b_2 = 2$	$b_3 = 3$	$b_4 = 1$	$\sum_{i=1}^{3} a_i = \sum_{j=1}^{4} b_j = 10$

It is easy to check that this solution is feasible and the fuel consumption is

$$F = \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} x_{ij} = 60$$

Next go to the next step and check the optimality of feasible solution.

2.3 Optimality condition

We will use the so called method of potentials, also known as "$u - \nu$" method . Consider auxiliary numbers $u_1, u_2, ..., u_k$ and $\nu_1, \nu_2, ..., \nu_l$. For any admissible solution the value $\sum_{i=1}^{k}\sum_{j=l}^{l}(u_i + \nu_j)x_{ij}$ is the same and constant:

$$\sum_{i=1}^{k}\sum_{j=1}^{l}(u_i + \nu_j)x_{ij} = \sum_{i=1}^{k} u_i \sum_{j=1}^{l} x_{ij} + \sum_{j=1}^{l} \nu_j \sum_{i=1}^{k} x_{ij} = \sum_{i=1}^{k} u_i a_i + \sum_{j=1}^{l} \nu_j b_j = C$$

Next, assume that for some admissible solution we found the numbers u_i and ν_j such that the following conditions

$$\begin{aligned} u_i + \nu_j &= c_{ij},\ for\ x_{ij} > 0, \\ u_i + \nu_j &\leq c_{ij},\ for\ x_{ij} = 0 \end{aligned} \tag{2.8}$$

hold.

The solution is called potential solution if it satisfies to condition (2.21) and the sum $u_i + \nu_j = \bar{c_{ij}}$ called pseudocost. Then the condition for potential solution can be rewritten (2.21) as

$$\begin{aligned} \bar{c_{ij}} - c_{ij} &= 0,\ for\ x_{ij} > 0, \\ \bar{c_{ij}} - c_{ij} &\leq 0,\ for\ x_{ij} = 0 \end{aligned} \tag{2.9}$$

The general cost of fuel consumption for potential solution is equal C. Indeed, we can replace c_{ij} in (8.1) by the sum $u_i + \nu_j$, since the component for which $u_i + \nu_j < c_{ij}$ are equal to zero. If change the potential solution in a such way that some positive components $x_{ij} > 0$ becomes zero and some zero components $x_{ij} = 0$ becomes positive (i.e. change basis), then new solution will be not a potential, while for some new $x_{ij}^{'} > 0$ we will have $u_i + \nu_j \leq c_{ij}$. The cost function for this new solution is

$$F^{'} = \sum_{i=1}^{k}\sum_{j=1}^{l} c_{ij}x_{ij}^{'} \geq \sum_{i=1}^{k}\sum_{j=1}^{l}(u_i + \nu_j)x_{ij}^{'} = C$$

or in other words

$$F^{'} \geq F.$$

Thus, the general cost of fuel consumption for any solution can not be less then for potential solution. It means that potential solution is optimal. In another words if we have even one component $x_{ij} = 0$ with $\bar{c_{ij}} - c_{ij} > 0$ then the solution is not optimal.

Let us check the optimality condition for our problem. Consider the following table

	B_1	B_2	B_3	B_4	a_i	u_i
A_1	$\bar{c_{11}} = 4$ $c_{11} = 4$ $x_{11} = 3$				3	
A_2	4 4 1	12 12 2	5 5 2		5	
A_3			6 6 1	4 4 1	2	
b_j	4	2	3	1	F=60	
ν_j						

Then we should find potential u_i and ν_j such that for $x_{ij} > 0$ the condition $c_{ij} = u_i + \nu_j$ hold. One of the potentials can be chosen arbitrary. Let $\nu_1 = 0$, since $u_1 + \nu_1 = 4$ then $u_1 = 4$. Next following this logic we found step by step:

$$u_2 = 4 \longrightarrow \nu_2 = 8 \longrightarrow \nu_3 = 1 \longrightarrow u_3 = 5 \longrightarrow \nu_4 = -1$$

and calculate $\bar{c_{ij}}$ for zero components $x_{ij} = 0$. Than we will have the following table:

	B_1	B_2	B_3	B_4	a_i	u_i
A_1	4 4 3	12 6	5 8	3 2	3	4
A_2	4 4 1	12 12 2	5 5 2	3 1	5	4
A_3	5 8	13 10	6 6 1	4 4 1	2	5
b_j	4	2	3	1		
ν_j	0	8	1	-1		

Now we are ready to check our initial supporting feasible solution for optimality. Namely to check the condition $\bar{c_{ij}} - c_{ij} \leq 0$ for $x_{ij} = 0$. In our case for several zero components of our feasible solution this conditions are not satisfied. Hence our solution is not optimal. In next section we will consider how to improve feasible solution.

2.4 Improvement of the feasible solution

In order to improve the feasible solution we should find the zero components for which the difference $\bar{c_{ij}} - c_{ij} > 0$ is maximal and to define the new value $\theta > 0$ for this component. Then we make a necessary corrections of the previous solution in order it remains feasible (i.e. the constraints of our problem should by satisfy).

	B_1	B_2	B_3	B_4	a_i	u_i
A_1	4 4 $3-\theta$	12 6 θ	5 8	3 2	3	4
A_2	4 4 $1+\theta$	12 12 $2-\theta$	5 5 2	3 1	5	4
A_3	5 8	13 10	6 6 1	4 4 1	2	5
b_j	4	2	3	1		
ν_j	0	8	1	-1		

Now we can define the value of the θ by replacement one of the positive components from previous solution by zero component, and also we should not have the negative value of x_{ij}. The maximal admissible value of θ to provide the condition above can be find from $\min(3-\theta; 2-\theta) = 0 \Longrightarrow \theta = 2$. Then we have the new feasible solution:

	B_1	B_2	B_3	B_4	a_i	u_i
A_1	4 4 1	6 6 2			3	
A_2	4 4 3		5 5 2		5	
A_3			6 6 1	4 4 1	2	
b_j	4	2	3	1	F=48	
ν_j						

Now, we are need to repeat the described procedure again, namely we will need to calculate new potentials: Let $\nu_1 = 0$, since $u_1 + \nu_1 = 4$ then $u_1 = 4$. Next

$$u_2 = 4 \longrightarrow \nu_2 = 2 \longrightarrow \nu_3 = 1 \longrightarrow u_3 = 5 \longrightarrow \nu_4 = -1.$$

And then new value $\bar{c_{ij}}$ for zero components of the solution.

	B_1	B_2	B_3	B_4	a_i	u_i
A_1	4 4 1	6 6 2	5 8	3 2	3	4
A_2	4 4 3	6 6	5 5 2	3 1	5	4
A_3	5 8	7 10	6 6 1	4 4 1	2	5
b_j	4	2	3	1		
ν_j	0	2	1	-1		

The optimality conditions are not holds for several zero components. Find the maximal value of $\bar{c_{ij}} - c_{ij}$. It is easy to check that maximal value located in A_2B_4.

	B_1	B_2	B_3	B_4	a_i	u_i
A_1	4 4 1	6 6 2	5 8	3 2	3	4
A_2	4 4 3	6 6	5 5 $2-\theta$	3 1 θ	5	4
A_3	5 8	7 10	6 6 $1+\theta$	4 4 $1-\theta$	2	5
b_j	4	2	3	1		
ν_j	0	2	1	-1		

Find the value $\theta = \min(2-\theta, 1-\theta) = 0 \longrightarrow \theta = 1$. Then we will have the following new feasible solution:

	B_1	B_2	B_3	B_4	a_i	u_i
A_1	4 4 1	6 6 2			3	
A_2	4 4 3		5 5 1	1 1 1	5	
A_3			6 6 2		2	
b_j	4	2	3	1	F=46	
ν_j						

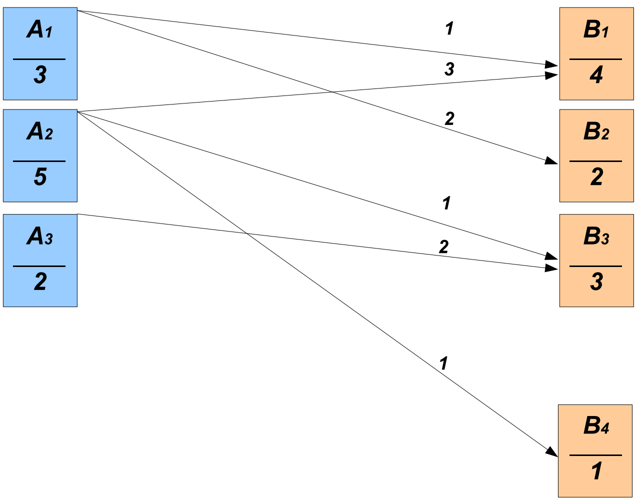
Let $\nu_1 = 0$, since $u_1 + \nu_1 = 4$ then $u_1 = 4$. Next

$$u_2 = 4 \longrightarrow \nu_2 = 2 \longrightarrow \nu_3 = 1 \longrightarrow u_3 = 5 \longrightarrow \nu_4 = -3.$$

And calculate the new value $\bar{c_{ij}}$ for zero components of the solution.

	B_1	B_2	B_3	B_4	a_i	u_i
A_1	4 4 1	6 6 2	5 8	1 2	3	4
A_2	4 4 3	6 12	5 5 1	1 1 1	5	4
A_3	5 8	7 10	6 6 2	2 4	2	5
b_j	4	2	3	1		
ν_j	0	2	1	-3		

It is easy to see that optimality condition are holds for all zero components of the feasible solution, that means this solution is optimal. Thus we will need to send our UAVs as follows:



To area of operation number one B_1: 1 UAV from aerobase A_1 and 3 UAVs from A_2;
To area of operation B_2: 2 UAVs from A_1;
To area of operation B_3: 1 UAV from A_2 and 2 UAVs from A_3;
To area of operation B_4: 1 UAV from A_2.

Remark This method as well as simplex methods are iterative, finite, exact(satisfied all constraints) and relaxed(in a sense of the value of objective function). Thus in some sense this method is analog of simplex method, but the ideas of this method is more naturally can be applied to assignments LP problems.

2.5 Illustrative example

Assume that we have 3 airbases locate at Changi A_1 with 3 UAVs $(a_1 = 3)$, Jurong West A_2 with 3 UAVs $(a_2 = 3)$, and Woodland A_3 with 1 UAV $(a_1 = 1)$. Now 7 UAVs are requested from B_1-Raffles Place $(b_1 = 2)$, B_2-Jurong Island $(b_2 = 2)$, and B_3- Sentosa Island $(b_3 = 3)$. The distances between A_i and B_j given below in kilometers:

	B_1	B_2	B_3
A_1	13	30	18
A_2	16	9	17
A_3	21	20	23

(2.10)

Table 1: Distances between aerobases A_i and area of operations B_j

The speed of UAVs are fixed $v_{ij} = 30\frac{m}{sec}$.

Next, for all i and j denote by $c_{ij} = \frac{d_{ij}}{v_{ij}}$ the benefit of sending the UAV from i-th aerobase to j-th zone of area of operation. The benefit means the flight time from $A_i \to B_j$.

	B_1	B_2	B_3
A_1	433	1000	600
A_2	533	300	566
A_3	700	666	766

(2.11)

Table 2: UAVs flight time from $A_i \to B_j$ (2.12)

Then using our notation we can formulate the problem statement as the following integer programming problem: To find $x_{ij}, (i = 1, 2, 3; j = 1, 2, 3)$ such that, the total service time performed by all UAVs takes a maximal value

$$T^{service} = \sum_{i=1}^{3} h_i - 2 \min_{x_{ij}} \sum_{i=1}^{3} \sum_{j=1}^{3} \frac{d_{ij}}{v_{ij}} x_{ij} \to \max_{x_{ij}} \tag{2.13}$$

Remark 1:

The service time for each UAVs is equal to their endurance h_i minus the time needed to reach the preassigned zone and come back to the base. Thus the total service time of the group of UAVs involved in the mission is given by (2.13). Hence the total service time of the group of UAVs involved in the mission will be maximal if the total flight time to reach the preassigned zones is minimal

$$F = \sum_{i=1}^{3}\sum_{j=1}^{3} t_{ij} = \sum_{i=1}^{3}\sum_{j=1}^{3} c_{ij}x_{ij} \to \min_{x_{ij}} \quad \text{where } c_{ij} = \frac{d_{ij}}{v_{ij}} \tag{2.14}$$

Then we can consider the following optimization problem:

$$F = \sum_{i=1}^{3}\sum_{j=1}^{3} c_{ij}x_{ij} \to \min_{x_{ij}} \tag{2.15}$$

$$\text{subject to} \tag{2.16}$$

$$\sum_{i=1}^{3} x_{ij} = b_j, \quad j = 1,2,3$$

$$\sum_{j=1}^{3} x_{ij} = a_i, \quad i = 1,2,3 \tag{2.17}$$

$$\sum_{i=1}^{3} a_i = \sum_{j=1}^{3} b_j \tag{2.18}$$

$$x_{ij} \geq 0, \quad x_{ij} \in \mathbb{N}. \tag{2.19}$$

where $A_i, i = 1,2,3$- number of aerobases,
$a_1 = 3, a_2 = 3, a_3 = 1$ - number of UAVs located in A_i,
$B_j, j = 1,2,3$- areas of operations,
$b_1 = 2, b_2 = 2, a_3 = 3$- numbers of UAVs for service of B_j
d_{ij}- distances from A_i to B_j given in table 1 , $v_{ij} = 30\frac{m}{sec}$
x_{ij}-number of UAVs from A_i to B_j
c_{ij}- given in table 2.

The first two constraints means that we are need to satisfy all service request from each area of operation and to use for that purpose all available UAVs. The last constraint means that the sum of all requests are equal to the total number of available UAVs.

The condition of that problem can be represented in table form:

	B_1	B_2	B_3	a_i
A_1	x_{11}	x_{12}	x_{13}	$a_1 = 3$
A_2	x_{21}	x_{22}	x_{23}	$a_2 = 3$
A_3	x_{31}	x_{32}	x_{33}	$a_3 = 1$
b_j	$b_1 = 2$	$b_2 = 2$	$b_3 = 3$	$\sum_{i=1}^{3} a_i = \sum_{j=1}^{3} b_j = 7$

Below we give the detailed step-by-step procedure to determine the optimal solution.

2.5.1 Initial feasible solution

To construct the initial feasible solution we will use "North-West corner" method. The construction of the initial supporting feasible solution consist from the several steps on each of them are filled either a row or a table column. The procedure begins with the left top ("northwest") element $x_{11} = \min(a_1; b_1)$ of the plan. If $a_1 < b_1$, i.e. $x_{11} = a_1$, than from the further consideration we eliminate all elements from the first row. If $a_1 \geq b_1$, i.e. $x_{11} = b_1$, than all elements from the first column are eliminated. In the case $a_1 < b_1$ the next element of feasible solution will be chosen from the second row by the rule $x_{21} = \min(a_2; b_1 - a_1)$. Next, if $a_2 < b_1$, i.e. $x_{21} = a_2$, and in this case we eliminated from our further consideration all elements from the second row. If $a_2 \geq b_1 - a_1$, i.e. $x_{21} = b_1 - a_1$, and further we will not consider the elements from the first column. The next assignment will be made on the intersection of the second column and second row as follows: $x_{22} = \min(a_1 + a_2 - b_1; b_2)$. Then repeated this procedure we will find all elements of the initial supporting feasible solution.

In our case we have the following:

$$\begin{aligned} x_{11} &= \min(a_1; b_1) = \min(3; 2) = 2 \\ \Downarrow & \\ x_{12} &= \min(a_1 - b_1; a_2) = \min(1; 2) = 1 \\ \Downarrow & \\ x_{22} &= \min(a_2; b_2 + b_1 - a_!) = \min(3; 1) = 1 \\ \Downarrow & \\ x_{23} &= \min(a_2 + a_1 - b_1 - b_2; b_3) = \min(2; 3) = 2 \\ \Downarrow & \\ x_{33} &= \min(a_3; b_3 - (a_1 + a_2 - b_1 - b_2)) = \min(1; 1) = 1 \end{aligned} \tag{2.20}$$

	B_1	B_2	B_3	a_i
A_1	$x_{11} = 2$	$x_{12} = 1$		$a_1 = 3$
A_2		$x_{22} = 1$	$x_{23} = 2$	$a_2 = 3$
A_3			$x_{33} = 1$	$a_3 = 1$
b_j	$b_1 = 2$	$b_2 = 2$	$b_3 = 3$	$\sum\limits_{i=1}^{3} a_i = \sum\limits_{j=1}^{3} b_j = 7$

Next calculate the value of our cost function:

$$F = 433 \times 2 + 1000 \times 1 + 300 \times 1 + 566 \times 2 + 766 \times 1 = 4061 \; seconds \approx 67.7 \; minutes$$

2.5.2 Optimality condition

We will use the so called method of potentials, also known as "$u - \nu$" method . Consider auxiliary numbers $u_1, u_2, ..., u_k$ and $\nu_1, \nu_2, ..., \nu_l$. For any admissible solution the value $\sum\limits_{i=1}^{k} \sum\limits_{j=l}^{l} (u_i + \nu_j) x_{ij}$ is the same and constant:

$$\sum_{i=1}^{k} \sum_{j=1}^{l} (u_i + \nu_j) x_{ij} = \sum_{i=1}^{k} u_i \sum_{j=1}^{l} x_{ij} + \sum_{j=1}^{l} \nu_j \sum_{i=1}^{k} x_{ij} = \sum_{i=1}^{k} u_i a_i + \sum_{j=1}^{l} \nu_j b_j = C$$

Next, assume that for some admissible solution we found the numbers u_i and ν_j such that the following conditions

$$u_i + \nu_j = c_{ij},\ for\ x_{ij} > 0, \qquad (2.21)$$
$$u_i + \nu_j \leq c_{ij},\ for\ x_{ij} = 0$$

hold.

The solution is called potential solution if it satisfies to condition (2.21) and the sum $u_i + \nu_j = \bar{c_{ij}}$ called pseudocost. Then the condition for potential solution can be rewritten (2.21) as

$$\bar{c_{ij}} - c_{ij} = 0,\ for\ x_{ij} > 0, \qquad (2.22)$$
$$\bar{c_{ij}} - c_{ij} \leq 0,\ for\ x_{ij} = 0$$

Let us check the optimality condition for our problem. Consider the following table

	B_1	B_2	B_3	a_i	u_i
A_1	$\bar{c_{11}} = 433$ $c_{11} = 433$ $x_{11} = 2$	1000 1000 1		3	
A_2		330 330 1	566 566 2	3	
A_3			766 766 1	1	
b_j	2	2	3	F=4064	
ν_j					

Then we should find potential u_i and ν_j such that for $x_{ij} > 0$ the condition $c_{ij} = u_i + \nu_j$ hold. One of the potentials can be chosen arbitrary.

Let $\nu_1 = 0$, since $u_1 + \nu_1 = 433$ then $u_1 = 433$. Next following this logic we found step by step:

$$\nu_2 + u_1 = 1000 \longrightarrow \nu_2 = 1000 - 433 = 567,$$

$$\nu_2 + u_2 = 300 \longrightarrow u_2 = 300 - 567 = -267,$$

$$u_2 + \nu_3 = 566 \longrightarrow \nu_3 = 566 + 267 = 833,$$

$$\nu_3 + u_2 = 766 \longrightarrow u_3 = 766 - 833 = -67$$

Than we will have the following table:

	B_1	B_2	B_3	a_i	u_i
A_1	$\bar{c_{11}} = 433$ $c_{11} = 433$ $x_{11} = 2$	1000 1000 1		3	433
A_2		330 330 1	566 566 2	3	-267
A_3			766 766 1	1	-67
b_j	2	2	3	F=4064	
ν_j	0	567	833		

Now we are ready to check our initial supporting feasible solution for optimality. Namely to check the condition $\bar{c_{ij}} - c_{ij} \leq 0$ for $x_{ij} = 0$.

$\bar{c_{ij}} = u_i + \nu_j \longrightarrow \bar{c_{21}} = -267$
$\bar{c_{13}} = 433 + 833 = 1266,$
$\bar{c_{31}} = 0 - 67 = -67,$
$\bar{c_{32}} = 567 - 67 = 500$

Then in matrix of estimates $\Delta = c_{ij} - \bar{c_{ij}} = \begin{pmatrix} 0 & 0 & -666 \\ 800 & 0 & 0 \\ 767 & 166 & 0 \end{pmatrix}$ find a minimal element $\Delta_{13} = -666 = \min_{i,j} \Delta_{ij}$.

In our case for one zero component of our feasible solution this conditions are not satisfied. Hence our solution is not optimal.

2.5.3 Improvement of the feasible solution

Change the initial feasible solution by adding the value *theta* to element x_{13} with some corrections of other elements too.

	B_1	B_2	B_3	a_i
A_1	2	$1-\theta$	θ	$a_1 = 3$
A_2		$1+\theta$	$2-\theta$	$a_2 = 3$
A_3			1	$a_3 = 1$
b_j	$b_1 = 2$	$b_2 = 2$	$b_3 = 3$	$\sum_{i=1}^{3} a_i = \sum_{j=1}^{3} b_j = 7$

Find the value $\theta = \min(2-\theta, 1-\theta) = 0 \longrightarrow \theta = 1$. Then we will have the following new feasible solution:

	B_1	B_2	B_3	a_i
A_1	2		1	$a_1 = 3$
A_2		2	1	$a_2 = 3$
A_3			1	$a_3 = 1$
b_j	$b_1 = 2$	$b_2 = 2$	$b_3 = 3$	$\sum_{i=1}^{3} a_i = \sum_{j=1}^{3} b_j = 7$

Next calculate the value of our cost function:

$$F = 433 \times 2 + 600 \times 1 + 300 \times 2 + 566 \times 1 + 766 \times 1 = 3398 \; seconds \approx 56.6 \; minutes$$

Now, we are need to repeat the described procedure again, namely we will need to calculate new potentials:

Let $\nu_1 = 0$, since $u_1 + \nu_1 = 433$ then $u_1 = 433$. Next following this logic we found step by step:

$$\nu_3 + u_1 = 600 \longrightarrow \nu_3 = 600 - 433 = 167,$$

$$\nu_3 + u_2 = 566 \longrightarrow u_2 = 567 - 167 = 389,$$

$$\nu_2 + u_2 = 300 \longrightarrow \nu_2 = 300 + 399 = -99,$$

$$\nu_3 + u_3 = 766 \longrightarrow u_3 = 766 - 167 = 599$$

Than we will have the following table:

	B_1	B_2	B_3	a_i	u_i
A_1	$\bar{c_{11}} = 433$ $c_{11} = 433$ $x_{11} = 2$		600 600 1	3	433
A_2		330 330 2	566 566 1	3	399
A_3			766 766 1	1	599
b_j	2	2	3	F=3398	
ν_j	0	-99	167		

Now we are ready to check our supporting feasible solution for optimality. Namely to check the condition $\bar{c_{ij}} - c_{ij} \leq 0$ for $x_{ij} = 0$.

$\bar{c_{ij}} = u_i + \nu_j \longrightarrow \bar{c_{12}} = u_1 + \nu_2 = 433 - 99 = 334$
$\bar{c_{21}} = 399 + 0 = 399,$
$\bar{c_{31}} = 599 + 0 = 599,$
$\bar{c_{32}} = 599 - 99 = 500$

Then in matrix of estimates $\Delta = c_{ij} - \bar{c_{ij}} = \begin{pmatrix} 0 & 666 & 0 \\ 134 & 0 & 0 \\ 101 & 166 & 0 \end{pmatrix}$

The optimality conditions are satisfied, since $\forall \Delta_{ij} \geq 0$.

Optimal solution are

$$\begin{aligned} &x_{11} = 2; \quad x_{13} = 1; \\ &x_{22} = 2; \quad x_{23} = 1; \\ &x_{33} = 1. \end{aligned} \tag{2.23}$$

Thus we will need to send our UAVs as follows: from Changi to Raffles Place: 2 UAVs;
from Changi to Sentosa Island: 1 UAV;
from Jurong West to Jurong Island: 2 UAVs;
from Jurong West to Sentosa Island : 1 UAV;
from Woodland to Sentosa Island: 1 UAV.

Chapter 3

Dynamical assignment of UAVs for multiple tasks

3.1 Problem statement

Let $[0, H]$ is the given period for service of $B_1, B_2, ..., B_j, ..., B_l$ zones of area of operation (tasks). It is assumed that each onetime service of each zone B_j requests includes at least b_j numbers of UAVs, $j = 1, \ldots, l$. Also, assume that we have k aerobases $A_1, A_2, ..., A_i, ..., A_k$ with $a_1, a_2, ..., a_i, ..., a_k$ number of homogenous UAVs, respectively.

The problem is to assign UAVs between areas of operations $B_j, j = 1, \ldots, l$ in a such way that the total service "profit" will be maximal. The different notions of "profit" will be introduced later.

3.1.1 Variables and constants

Divide the interval $[0, H]$ by the moments $t_s = s\Delta,\ s = 1, 2, ..., \nu$ where $\nu = \left[\dfrac{H}{\Delta}\right]$ denotes the integer part of the fraction $\dfrac{H}{\Delta}$, and Δ is a small number. The concrete value of Δ can be stated experimentally and depends on efficiency of the used numerical algorithms.

Hence, we have the time interval partition

$$0 < \Delta < 2\Delta < ... < s\Delta < (s + 1)\Delta < .... < H.$$

For each discrete moment $t_s = s\Delta$, $s = 1, 2, ..., \nu$, introduce the following variables:

1. $x_{ij}(t_s)$ is the number of UAVs from i-th aerobase send to j-th zone at the moment t_s;

2. $a_i(t_s)$ is the number of UAVs at $i-th$ aerobase at the moment t_s;

3. $b_j(t_s)$ is the number of UAVs that are serving the $j-th$ zone at the moment t_s;

4. t_{ij} is the flight time from $i-th$ aerobase to $j-th$ zone;

5. k and l are the number of aerobases and zones for service, respectively;

6. h_i is the flight endurance for UAVs from $i-th$ aerobase.

Note that homogeneous of UAVs in each aerobase is not restricted since UAVs can be classified or, in final, in the simplest case we can consider the position when each aerobase is complicated by a single UAV. Obviously, at the initial moment $t = 0$ we have $b_j(0) = b_j,\ j = 1, 2, ..., l;\quad a_i(0) = a_i,\ i = 1, 2, ..., k.$

Now, we state the relation describing the dynamic of introduced variables.

3.1.2 Constraints

1) The number of UAVs at $i-th$ aerobase at the next moment $t_s+\Delta$ is composed of UAVs that are being at the previous moment t_s, plus UAVs that are returned during the period $[t_s, t_s+\Delta]$, and minus UAVs that were send to zones at the moment t_s. These facts give the following equalities

$$a_i(t_s+\Delta) = a_i(t_s) - \sum_{j=1}^{l} x_{ij}(t_s) + \sum_{j=1}^{l} x_{ij}(t_s+\Delta-h_i),\ i = 1, ..., k. \tag{3.1}$$

The term $\sum\limits_{j=1}^{l} x_{ij}(t_s+\Delta-h_i)$ denotes UAVs that were send early, and that should come back due to their flight endurance. Otherwise, the term $x_{ij}(t_s+\Delta-h_i)$ means that $i-th$ UAV has sufficient endurance to continue service of $j-th$ zone, and hence, it can not come back to aerobase. Here we consider those objects where argument $t_s+\Delta-h_i > 0$. The initial conditions are $a_i(0) = a_i,\ i = 1, 2, ..., k.$

2) The number of UAVs that will serve the $j-th$ zone at the next moment $t_s+\Delta$ is composed of UAVs that are serving this zone at the previous moment t_s and having sufficient flight endurance, plus UAVs that reach this zone during the period $(t_s, t_s+\Delta]$,

and minus UAVs that are out-of-fuel to the moment t. These facts lead the following equalities

$$b_j(t_s+\Delta) = b_j(t_s) - \sum_{i=1}^{k} x_{ij}(t_s - h_i + t_{ij}) + \sum_{i=1}^{k} x_{ij}(t_s - t_{ij}),\ j = 1, ..., l. \tag{3.2}$$

The term $\sum_{i=1}^{k} x_{ij}(t_s - h_i + t_{ij})$ denotes UAVs that should leave the $j-th$ zone due to their out-of-fuel. The term $\sum_{i=1}^{k} x_{ij}(t_s - t_{ij})$ denotes UAVs that were send early and should reach the $j-th$ zone during the period $(t_s, t_s+\Delta]$. Here we consider those objects where arguments $t_s - h_i + t_{ij} > 0$ and $t_s - t_{ij} > 0$. The initial conditions are $b_j(0) = b_j,\ j = 1, 2, ..., l$.

3) The variables $x_{ij}(t_s)$ at each moment $t_s, s = 1, ..., \nu$ satisfy the following conditions

$$\begin{aligned} a_i(t_s) + \sum_{j=1}^{l} x_{ij}(t_s) &= a_i,\ i = 1, ..., k. \\ b_j(t_s) + \sum_{i=1}^{k} x_{ij}(t_s - t_{ij}) &= b_j,\ (t_s - t_{ij} > 0)\ j = 1, ..., l. \end{aligned} \tag{3.3}$$

The first equation images the fact that the being UAVs can be allocated among zones. The second equation means that at each moment the service request should be satisfied.

The given above maim body of the problem constraints can be completed by additional conditions (constraints) followed from description of the Task 5.

4) Let τ_j^{first} is the given earliest time of 1-st visit to j zone for each $j,\ 1 \le j \le l$. Then the constraints (3.47)—(3.49) can be supplemented by the following:

$$\sum_{i=1}^{k} x_{ij}(s_j^{first}) \neq 0,\ 1 \le j \le l \tag{3.4}$$

where s_j^{first} is the discrete moment from the set $s = 1, 2, ..., \nu$ satisfying the following conditions:

$$s_j^{first}\Delta \le \tau_j^{first} \le (s_j^{first}+1)\Delta \qquad \text{for some} \ \ s_j^{first} \in \{1, 2, ..., \nu\}.$$

The inequality (3.50) means that there exist at least one aerobase such that their UAVs will start with 1-st service visit to j zone no later on the preassigned moment τ_j^{first}.

Remark. If the τ_j^{first} is treated as the moment before of which the service of j zone is prohibited, then the constraints (3.47)—(3.49) can be supplemented by the following:

$$x_{ij}(s\Delta) = 0,\ 1 \le j \le l,\ \forall\ s\Delta \le \tau_j^{first} \ \text{and} \ \ \forall\ i = 1, 2, ..., k \tag{3.5}$$

5) Let τ_j^{latest} is the given latest time of 1-st visit to j zone for each $j,\ 1 \le j \le l$. Then the constraints (3.47)—(3.49) can be supplemented by the following:

$$\sum_{i=1}^{k} x_{ij}(s_j^{first}) \neq 0, j,\ 1 \le j \le l \tag{3.6}$$

where s_j^{first} is the discrete moment from the set $s = 1, 2, ..., \nu$ satisfying the following conditions:

$$s_j^{first}\Delta \le \tau_j^{first} \le (s_j^{first} + 1)\Delta \qquad \text{for some} \quad s_j^{first} \in \{1, 2, ..., \nu\}$$

and such that

$$s_j^{first} + h_i \le \tau_j^{latest}, \quad 1 \le j \le l,\ \forall\ i \in I_j^{first} \tag{3.7}$$

where

$$I_j^{first} = \{i,\ 1 \le i \le k : x_{ij}(s_j^{first}) \neq 0\}.$$

The couple of inequalities (3.52) — (3.53) means that there exist at least one aerobase such that their UAV the 1-st visit to j zone will begin no later the pre-assigned earliest time τ_j^{first}, and the ending this 1-st visit to j zone will be no later the pre-assigned the latest time τ_j^{latest} .

6) Let τ_j^{last} is the given last time of visits to j zone for each $j,\ 1 \le j \le l$. Then the constraints (3.47)—(3.49) can be supplemented by the following:

$$x_{ij}(s_j^{last}) = 0, \quad \text{for all} \quad s_j^{last} \le t_s \le \nu, \quad 1 \le i \le k \quad \text{and} \quad 1 \le j \le l \tag{3.8}$$

where s_j^{last} is the discrete moment from the set $s = 1, 2, ..., \nu$ satisfying the following conditions:

$$(s_j^{last} - 1)\Delta \le \tau_j^{last} \le s_j^{last}\Delta \qquad \text{for some} \quad s_j^{last} \in \{1, 2, ..., \nu\}.$$

The equalities (3.54) denotes that all visits to j zone after the preassigned moment τ_j^{last} are prohibited.

3.1.3 Types of objective function

The cost value function used for optimization problem can be determined as follows:

a) the total service time for multiple zones

$$J_1(x) = \sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{s=0}^{\nu} x_{ij}(t_s)(h_i - 2t_{ij}). \tag{3.9}$$

b) the total number of UAVs "circles"

$$J_2(x) = \sum_{i=1}^{k} \sum_{j=1}^{l} \sum_{s=0}^{\nu} x_{ij}(t_s) \tag{3.10}$$

c) the total unobservable time for multiple zones

$$J_3(x) = \sum_{i=1}^{k} \sum_{j=1}^{l} \sum_{s=0}^{\nu} x_{ij}(t_s)(H - h_i - 2t_{ij}) \tag{3.11}$$

d) time of the first visit in the worst-case zone

$$J_4(x) = \max_{1 \le j \le k} t_j^{first}, \quad \text{where } t_j^{first} = \min_{1 \le i \le k} \min_{1 \le s \le \nu} \{t_s : x_{ij}(t_s) \ne 0.\} \tag{3.12}$$

Hence, the optimization problem can be equipped by any of the proposed cost functions. In addition, some combinations of these function with the proper weighting coefficients can be used as a new cost function.

Thus, the optimal schedule problem of UAVs for multiple zones can be formulated, for example in the case of maximization of the total service time for multiple zones, as the following special integer dynamical linear programming problem (we change t_s by $t_s = s\Delta,\ s = 1, 2, ..., \nu$): maximize the cost value function

$$J_1(x) = \sum_{i=1}^{k} \sum_{j=1}^{l} \sum_{s=0}^{\nu} x_{ij}(s\Delta)(h_i - 2t_{ij}) \to \max_{x_{ij}(s\Delta) \in \mathbb{N}, s=0,1,....,\nu} \tag{3.13}$$

subject to

$$\begin{aligned} a_i(s\Delta + \Delta) = a_i(s\Delta) - \sum_{j=1}^{l} x_{ij}(s\Delta) + \sum_{j=1}^{l} x_{ij}(s\Delta + \Delta - h_i),\ i = 1, ..., k. \\ b_j(s\Delta + \Delta) = b_j(s\Delta) - \sum_{i=1}^{k} x_{ij}(s\Delta - h_i + t_{ij}) + \sum_{i=1}^{k} x_{ij}(s\Delta - t_{ij}),\ j = 1, ..., l. \\ a_i(s\Delta) + \sum_{j=1}^{l} x_{ij}(s\Delta) = a_i,\ i = 1, ..., k. \\ b_j(s\Delta) + \sum_{i=1}^{k} x_{ij}(s\Delta - t_{ij}) = b_j,\ j = 1, ..., l, \\ s = 1, 2, ..., \nu, \end{aligned} \tag{3.14}$$

and

$$\sum_{i=1}^{k} x_{ij}(s_j^{first}) \ne 0,\ 1 \le j \le l \tag{3.15}$$

$$s_j^{first} + h_i \le \tau_j^{latest}, \quad 1 \le j \le l,\ \forall\ i \in I_j^{first}, \tag{3.16}$$

$$x_{ij}(s_j^{last}) = 0, \quad \text{for all} \quad s_j^{last} \le t_s \le \nu, \quad 1 \le i \le k \quad \text{and} \quad 1 \le j \le l \tag{3.17}$$

where

s_j^{first} is the discrete moment from the set $s = 1, 2, ..., \nu$ satisfying the conditions

$$s_j^{first}\Delta \leq \tau_j^{first} \leq (s_j^{first} + 1)\Delta \qquad \text{for some} \quad s_j^{first} \in \{1, 2, ..., \nu\},$$

s_j^{last} is the discrete moment from the set $s = 1, 2, ..., \nu$ satisfying the conditions:

$$(s_j^{last} - 1)\Delta \leq \tau_j^{last} \leq s_j^{last}\Delta \qquad \text{for some} \quad s_j^{last} \in \{1, 2, ..., \nu\}$$

and

$$I_j^{first} = \{i, \ 1 \leq i \leq k : x_{ij}(s_j^{first}) \neq 0\}.$$

Again note that $\nu = \left[\frac{H}{\Delta}\right]$ means the integer part of the fraction $\frac{H}{\Delta}$.

In (3.60) we consider those terms and elements where arguments $s\Delta - h_i + t_{ij} > 0$ and $s\Delta - t_{ij} > 0$.

Other optimization problem with another cost function mentioned above can be formulated by similar manner.

Remark 2.

The proposed partition of the planing horizon $[0, H]$ with small step Δ yields an ability to produce optimal schedule for UAVs, in fact, in regime of real time. The realization of this idea demands the development of some fast numerical algorithms for solution of the special classes of linear programming problems. Some new approaches to accelerate the solution of general linear programming problem is discussed in the paper [**?**]

Remark 3. In order to take into account the other request followed from description of the Task 5, the proposed model can be reformulated with the corresponding cost function. For example, the request to organize the zone service with maximum intervals between visits can be presented by maximization of the total unobservable time

$$J_3(x) = \sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{s=0}^{\nu} x_{ij}(t_s)(H - h_i - 2t_{ij}) \to \max_{x_{ij}(s\Delta)\in\mathbb{N}, s=0,1,...,\nu} \tag{3.18}$$

subject to constraints of (3.60) and (3.61).

The request to organize the zone service with minimum duration per visits can be presented by minimization of the total service time

$$J_1(x) = \sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{s=0}^{\nu} x_{ij}(t_s)(h_i - 2t_{ij}) \to \min_{x_{ij}(s\Delta)\in\mathbb{N}, s=0,1,...,\nu}. \tag{3.19}$$

subject to constraints of (3.60)and (3.61).

Remark 4. Another way to satisfy the multiple requests for zone service can be realized by optimization of one of the selected cost function and including the remained cost function into the main body of constraints (3.60)and (3.61) as follows, for example:

minimize the total number of UAVs "circles"

$$J_2(x) = \sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{s=0}^{\nu} x_{ij}(t_s) \to \min_{x_{ij}(s\Delta)\in\mathbb{N}, s=0,1,...,\nu}. \tag{3.20}$$

subject to constraints of (3.60)and (3.61) and the following new constraints

$$\sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{s=0}^{\nu} x_{ij}(s\Delta)(h_i - 2t_{ij}) \le A, \tag{3.21}$$

$$\sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{s=0}^{\nu} x_{ij}(t_s)(H - h_i - 2t_{ij}) \le B \tag{3.22}$$

where A and B are the known numbers those values can be given by specialists or can be determined experimentally.

Solution result presentation

The obtained solution of optimization problem (3.59)—(3.61) can be presented in the form convenient for the practical uses. Such presentation can be done, for example, by Diagram or Schedule Table of time and duration visits by each MAS for the chosen zones.

Let $x^0_{ij}(t_s),\ s = 1, ..., \nu,\ i = 1, ..., k,\ j = 1, ..., l$ is optimal solution of (3.59)— (3.61) where $t_s = s\Delta$ and Δ is the sampling (discretisation) step.

First, we indicate resulting information concerning history of zone observation due to the obtained solution. For each zone j, where $1 \le j \le l$, define the following characteristics:

- $I_j = \{i \in \{1, 2, ..., k\} : x^0_{ij}(t_s) \ne 0, s = 1, 2..., \nu\}$ — the indexes of aerobase used for observation of j zone;
- $k_j = |I_j|$—the total number of aerobases involved in observation of j zone. Here $|I_j|$ denotes the number of elements of the set I_j);
- $m_j = |S_j|,$ where $S_j = \{s \in \{1, 2, ..., \nu\} : x^0_{ij}(t_s) \ne 0, i = 1, 2..., k\}$ — the number of the discrete intervals of the form $[t_s, t_s + \Delta]$ during of which the observation by UAVs is realized for j zone. (Here $|S_j|$ denotes the number of elements of the set S_j);

- $T_j^{zone} = m_j\Delta$— the total duration of observation time for the j zone;
- $N_j^{zone} = \sum\limits_{i=1}^{k}\sum\limits_{s=1}^{\nu} x_{ij}^0(t_s)$—total number of UAVs used for observation of j zone;
- $t_j^{first} = \min\limits_{1\le s\le\nu}\{t_s : x_{ij}^0(t_s) \ne 0, i = 1, 2..., k\}$—the time of first visit to j zone;
- $t_j^{last} = \max\limits_{1\le s\le\nu}\{t_s : x_{ij}^0(t_s) \ne 0, i = 1, 2..., k\} + \Delta$—the time of the ending of observation of j zone;
- i_j^{first} —- those aerobases the UAVs of which were the first visitors of j zone, where i_j^{first} is the indexes from the set $\{1, 2, ..., k\}$ where the minimum for t_j^{first} is reached ;
- i_j^{last} — those aerobases the UAVs of which were the last visitors of j zone, where i_j^{last} is the indexes from the set $\{1, 2, ..., k\}$ where the maximum for the value t_j^{last} is reached.

It is possible, also, to restore more detail information concerning the visits of j zone between the first moment t_j^{first} of visit and the last moment t_j^{last} with detailed history for each observation intervals and UAVs involved in this observation.

On other hand, for UAVs addressed to observation mission can be useful information concerning their visits schedule for all zones for observation of which they are used.

For each i aerobase, where $1 \le i \le k$, define the following characteristics:

- $J_i = \{j \in \{1, 2, ..., l\} : x_{ij}^0(t_s) \ne 0, s = 1, 2..., \nu\}$ — the indexes of zone for observation of which the i aerobase is used;

  $M_i = \sum\limits_{i=1}^{k}\sum\limits_{s=1}^{\nu} x_{ij}^0(t_s)$—total number of UAVs used for observation of j zone;
- $\tau_i^{first} = \min\limits_{1\le s\le\nu}\{t_s : x_{ij}^0(t_s) \ne 0, j = 1, 2..., l\}$—the time of the first mission fly of UAVs of i aerobase;
- j_i^{first} —- those zones, for which the UAVs of i aerobase are used firstly for observation mission, where j_i^{first} is the indexes from the set $\{1, 2, ..., l\}$ where the minimum for τ_i^{first} is reached ;

This list of characteristics for UAVs of i aerobase can be continued by obviously manner.

3.1.4 Reformulation in matrix form

The proposed dynamical transportation problem (3.59)—(3.61) for allocation of MAS can be presented as a statistic problem given in the previous paragraph. But this way leads to the huge dimensions of the variables involved, and this together the specific structure of the considered problem are a serious obstacle for suitable solution for reasonable time. By this reason the development of special methods and design on this base of fast numerical methods for assignment problems of MAS with next their realization in the corresponding computer chips are actual and will be done at this work.

Introduce the following matrixes

$$A_{k\times\nu}=\begin{pmatrix} a_1(\Delta) & a_1(2\Delta) & ... & a_1(\nu\Delta)\\ a_2(\Delta) & a_2(2\Delta) & ... & a_2(\nu\Delta)\\ a_3(\Delta) & a_3(2\Delta) & ... & a_3(\nu\Delta)\\ ... & ... & ... & ...\\ a_k(\Delta) & a_k(2\Delta) & ... & a_k(\nu\Delta)\end{pmatrix}, \tag{3.23}$$

$$H^-_{\nu\times(\nu-1)}=\begin{pmatrix} 0 & 0 & 0 & ... & 0\\ 1 & 0 & 0 & ... & 0\\ 0 & 1 & 0 & ... & 0\\ ... & ... & ... & ... & ...\\ 0 & 0 & 0 & ... & 1\end{pmatrix}, \tag{3.24}$$

$$H^+_{\nu\times(\nu-1)}=\begin{pmatrix} 1 & 0 & 0 & ... & 0\\ 0 & 1 & 0 & ... & 0\\ ... & ... & ... & ... & ...\\ 0 & 0 & 0 & ... & 1\\ 0 & 0 & 0 & ... & 0\end{pmatrix}, \tag{3.25}$$

$$X_i=\begin{pmatrix} x_{i1}(\Delta) & x_{i1}(2\Delta) & ... & x_{i1}(\nu\Delta)\\ x_{i2}(\Delta) & x_{i2}(2\Delta) & ... & x_{i2}(\nu\Delta)\\ ... & ... & ... & ...\\ x_{il}(\Delta) & x_{il}(2\Delta) & ... & x_{il}(\nu\Delta)\end{pmatrix}_{l\times\nu}, i=1,...,k \tag{3.26}$$

Introduce the block matrixes of the form

$$X=\begin{pmatrix} X_1\\ X_2\\ ...\\ X_k\end{pmatrix}_{kl\times\nu},\ \Pi=\begin{pmatrix} e_l & 0_l & ... & 0_l\\ 0_l & e_l & ... & 0_l\\ ... & ... & ... & ...\\ 0_l & 0_l & ... & e_l\end{pmatrix}_{k\times kl} \tag{3.27}$$

where

$$e_l=\begin{pmatrix} 1 & 1 & ... & 1\end{pmatrix}_{1\times l}, 0_l=\begin{pmatrix} 0 & 0 & ... & 0\end{pmatrix}_{1\times l}, \tag{3.28}$$

Remark. Since the unknown variables of the optimization problem are $x_{ij}(\Delta), x_{ij}(2\Delta), ..., x_{ij}(\nu\Delta)$, then the other variables $x_{ij}(t)$ with argument t that is not coincide with arguments $\Delta, 2\Delta, ..., \nu\Delta$ will be approximated by the variables $x_{ij}(s\Delta)$ where $s = \left[\dfrac{t}{\Delta}\right]$ is the integer part of the number $s = \dfrac{t}{\Delta}$ such that the argument $s\Delta$ is the nearest to the argument t. Such kind approximation is admissible due to the freedom in choice of sampling step Δ. We assume, in fact, that for the considered optimization problem the unknown continuous function $x_{ij}(\tau)$ of the real variable τ can be approximated by piecewise constant function $x_{ij}(s\Delta),\ s = 1, ..., \nu$.

Noting the given remark, introduce the following matrixes

$$
\begin{aligned}
h(X_i) &= \begin{pmatrix}
x_{i1}\left(\Delta\left[\frac{\Delta - h_i}{\Delta}\right]\right) & x_{i1}\left(\Delta\left[\frac{2\Delta - h_i}{\Delta}\right]\right) & ... & x_{i1}\left(\Delta\left[\frac{\nu\Delta - h_i}{\Delta}\right]\right) \\
x_{i2}\left(\Delta\left[\frac{\Delta - h_i}{\Delta}\right]\right) & x_{i2}\left(\Delta\left[\frac{2\Delta - h_i}{\Delta}\right]\right) & ... & x_{i2}\left(\Delta\left[\frac{\nu\Delta - h_i}{\Delta}\right]\right) \\
... & ... & ... & ... \\
x_{il}\left(\Delta\left[\frac{\Delta - h_i}{\Delta}\right]\right) & x_{il}\left(\Delta\left[\frac{2\Delta - h_i}{\Delta}\right]\right) & ... & x_{il}\left(\Delta\left[\frac{\nu\Delta - h_i}{\Delta}\right]\right)
\end{pmatrix}_{l\times\nu}, \\
i &= 1, ..., k
\end{aligned}
$$

Introduce the block matrixes of the form

$$
h(X) = \begin{pmatrix} h(X_1)H^- \\ h(X_2)H^- \\ ... \\ h(X_k)H^- \end{pmatrix}_{kl\times(\nu)}, \ a_{k\times1} = \begin{pmatrix} a_1 \\ a_2 \\ ... \\ a_k \end{pmatrix}, e_{1\times\nu} \begin{pmatrix} 1 & 1 & ... & 1 \end{pmatrix} \tag{3.29}
$$

Then the first and third equations of (3.60) can be written in the matrix form as follows

$$
AH^- = AH^+ - \Pi X + \Pi h(X) \tag{3.30}
$$

$$
A + \Pi X = a_{k\times1}e_{1\times\nu} \tag{3.31}
$$

In order to rewrite the remained equations of (3.60) introduce the matrixes

$$B = \begin{pmatrix} b_1(\Delta) & b_1(2\Delta) & ... & b_1(\nu\Delta) \\ b_2(\Delta) & b_2(2\Delta) & ... & b_2(\nu\Delta) \\ b_3(\Delta) & b_3(2\Delta) & ... & b_3(\nu\Delta) \\ ... & ... & ... & ... \\ b_l(\Delta) & b_l(2\Delta) & ... & b_l(\nu\Delta) \end{pmatrix}_{l\times\nu}, \quad (3.32)$$

$$T(X_i) = \begin{pmatrix} x_{i1}\Big(\Delta\Big[\frac{\Delta - t_{i1}}{\Delta}\Big]\Big) & x_{i1}\Big(\Delta\Big[\frac{2\Delta - t_{i1}}{\Delta}\Big]\Big) & ... & x_{i1}\Big(\Delta\Big[\frac{\nu\Delta - t_{i1}}{\Delta}\Big]\Big) \\ x_{i2}\Big(\Delta\Big[\frac{\Delta - t_{i1}}{\Delta}\Big]\Big) & x_{i2}\Big(\Delta\Big[\frac{2\Delta - t_{i1}}{\Delta}\Big]\Big) & ... & x_{i2}\Big(\Delta\Big[\frac{\nu\Delta - t_{i1}}{\Delta}\Big]\Big) \\ ... & ... & ... & ... \\ x_{il}\Big(\Delta\Big[\frac{\Delta - t_{i1}}{\Delta}\Big]\Big) & x_{il}\Big(\Delta\Big[\frac{2\Delta - t_{i1}}{\Delta}\Big]\Big) & ... & x_{il}\Big(\Delta\Big[\frac{\nu\Delta - t_{i1}}{\Delta}\Big]\Big) \end{pmatrix}_{l\times\nu}, \quad (3.33)$$

$$TH(X_i) = \begin{pmatrix} x_{i1}\Big(\Delta\Big[\frac{\Delta - h_i + t_{i1}}{\Delta}\Big]\Big) & x_{i1}\Big(\Delta\Big[\frac{2\Delta - h_i + t_{i1}}{\Delta}\Big]\Big) & ... & x_{i1}\Big(\Delta\Big[\frac{\nu\Delta - h_i + t_{i1}}{\Delta}\Big]\Big) \\ x_{i2}\Big(\Delta\Big[\frac{\Delta - h_i + t_{i2}}{\Delta}\Big]\Big) & x_{i2}\Big(\Delta\Big[\frac{2\Delta - h_i + t_{i2}}{\Delta}\Big]\Big) & ... & x_{i2}\Big(\Delta\Big[\frac{\nu\Delta - h_i + t_{i2}}{\Delta}\Big]\Big) \\ ... & ... & ... & ... \\ x_{il}\Big(\Delta\Big[\frac{\Delta - h_i + t_{il}}{\Delta}\Big]\Big) & x_{il}\Big(\Delta\Big[\frac{2\Delta - h_i + t_{il}}{\Delta}\Big]\Big) & ... & x_{il}\Big(\Delta\Big[\frac{\nu\Delta - h_i + t_{il}}{\Delta}\Big]\Big) \end{pmatrix}_{l\times\nu} \quad (3.34)$$

$$i = 1, ..., k \quad (3.35)$$

and

$$T(X) = \begin{pmatrix} T(X_1) \\ T(X_2) \\ ... \\ T(X_k) \end{pmatrix}_{lk\times\nu}, \; TH(X) = \begin{pmatrix} TH(X_1) \\ TH(X_2) \\ ... \\ TH(X_k) \end{pmatrix}_{lk\times\nu} \quad (3.36)$$

Then the second an forth equations of (3.60) can be written as

$$BH^- = BH^+ - \Pi TH(X) + \Pi T(X), \quad (3.37)$$

$$B + \Pi T(X) = b_{l\times 1}e_{1\times\nu} \quad (3.38)$$

where

$$b_{l\times 1} = \begin{pmatrix} b_1 \\ b_2 \\ ... \\ b_l \end{pmatrix}, e_{1\times\nu}\begin{pmatrix} 1 & 1 & ... & 1 \end{pmatrix}$$

Finally, for example, the cost function $J_2(x) = \sum\limits_{i=1}^{k}\sum\limits_{j=1}^{l}\sum\limits_{s=1}^{\nu} x_{ij}(t_s)$ can be written as

$$J_2(X) = e_{kl}^T X e_{kl} \tag{3.39}$$

where $e_{kl}^T = \begin{pmatrix} 1 & 1 & ... & 1 \end{pmatrix}_{1\times kl}$ is the unit vector.

Thus, the matrix optimization problem is to find the integer valued matrix X maximizing the cost function

$$J_2(X) = e_{kl}^T X e_{kl} \to \max_X \tag{3.40}$$

subject to

$$\begin{aligned} A(H^- - H^+) &= \Pi\left(h(X) - X\right) && (3.41)\\ A - a_{k\times 1}e_{1\times\nu} &= -\Pi X && (3.42)\\ B(H^- - H^+) &= \Pi\left(T(X) + TH(X)\right), && (3.43)\\ B - b_{l\times 1}e_{1\times\nu} &= -\Pi T(X) && (3.44)\end{aligned}$$

This problem can be rewritten in the coordinate form, also.

3.2 Specific problem statement

3.2.1 Notation

$A_i, i = 1, ..., k$- aerobases,
a_i - number of UAVs located in A_i,
$B_j, j = 1, ..., l$- areas of operations,
b_j- numbers of UAVs for service of B_j
d_{ij}- distance from A_i to B_j,
v_{ij}- UAVs speed
x_{ij}-number of UAVs from A_i to B_j,
h_i- UAVs endurance located on A_i aerobase

3.2.2 Problem statement

$$T^{service} = \sum_{i=1}^{k} h_i - 2\sum_{i=1}^{k}\sum_{j=1}^{l}\frac{d_{ij}}{v_{ij}}x_{ij} \to \max_{x_{ij}} \tag{3.45}$$

subject to

$$\begin{aligned}
\sum_{i=1}^{k} x_{ij} = b_j, \quad j = 1, 2, ..., l \\
\sum_{j=1}^{l} x_{ij} = a_i, \quad i = 1, 2, ..., k \\
\sum_{i=1}^{k} a_i = \sum_{j=1}^{l} b_j \\
\frac{d_{ij}}{v_{ij}} \operatorname{sign}(x_{ij}) \geq t_j^{first}, \ i = 1, 2, ..., k \\
\frac{d_{ij}}{v_{ij}} \operatorname{sign}(x_{ij}) \leq t_j^{last}, \ i = 1, 2, ..., k \\
x_{ij} \geq 0, \qquad x_{ij} \ \text{ are integer numbers.}
\end{aligned} \tag{3.46}$$

Here the cost function presents the total service time performed by all UAVs used for mission. The first two constraints means that we are need to satisfy all service request from each area of operation and to use for that purpose all available UAVs. The last inequalities image the fact that the period of the start of service of $j-th$ zone by $i-th$ UAV is restricted by the pre-assigned time interval $[t_j^{first}, t_j^{last}]$. The cost function presents the total service time performed by all UAVs used for mission.

3.3 Dynamical assignment of UAVs with timing constraints

The optimal timing of air-to-zone (area of operation) tasks is undertaken. specifically, a scenario where multiple airvacles, located at airbases are required to prosecute geographically dispersed zones is considered.

3.3.1 Problem statement

Let $[0, H]$ is the given period for service of $B_1, B_2, ..., B_j, ..., B_l$ zones of area of operation. It is assumed that each onetime service of each zone B_j requests includes at least b_j numbers of UAVs, $j = 1, \ldots, l$. Also, assume that we have k aerobases $A_1, A_2, ..., A_i, ..., A_k$ with $a_1, a_2, ..., a_i, ..., a_k$ number of homogenous UAVs, respectively.

The problem is to assign UAVs between areas of operations $B_j, j = 1, \ldots, l$ in a such way that the total service "profit" will be maximal, and taking into account a "time windows" requirements. The different notions of "profit" will be introduced later.

3.3.2 Variables and constants

Divide the interval $[0, H]$ by the moments $t_s = s\Delta,\ s = 1, 2, \ldots.., \nu$ where $\nu = \left[\dfrac{H}{\Delta}\right]$ denotes the integer part of the fraction $\dfrac{H}{\Delta}$, and Δ is a small number. The concrete value of Δ can be stated experimentally and depends on efficiency of the used numerical algorithms.

Hence, we have the time interval partition

$$0 < \Delta < 2\Delta < \ldots < s\Delta < (s+1)\Delta < \ldots. < H.$$

For each discrete moment $t_s = s\Delta$, $s = 1, 2, \ldots, \nu$, introduce the following variables:

1. $x_{ij}(t_s)$ is the number of UAVs from i-th aerobase send to j-th zone at the moment t_s;
2. $a_i(t_s)$ is the number of UAVs at $i - th$ aerobase at the moment t_s;
3. $b_j(t_s)$ is the number of UAVs that are serving the $j - th$ zone at the moment t_s;
4. t_{ij} is the flight time from $i - th$ aerobase to $j - th$ zone;
5. k and l are the number of aerobases and zones for service, respectively;
6. h_i is the flight endurance for UAVs from $i - th$ aerobase.

Note that homogeneous of UAVs in each aerobase is not restricted since UAVs can be classified or, in final, in the simplest case we can consider the position when each aerobase is complicated by a single UAV. Obviously, at the initial moment $t = 0$ we have $b_j(0) = b_j,\ j = 1, 2, \ldots., l;\ \ a_i(0) = a_i,\ i = 1, 2, \ldots, k$.

Now, we state the relation describing the dynamic of introduced variables.

3.3.3 Constraints

1) The number of UAVs at $i - th$ aerobase at the next moment $t_s + \Delta$ is composed of UAVs that are being at the previous moment t_s, plus UAVs that are returned during the period

$[t_s, t_s + \Delta]$, and minus UAVs that were send to zones at the moment t_s. These facts give the following equalities

$$a_i(t_s + \Delta) = a_i(t_s) - \sum_{j=1}^{l} x_{ij}(t_s) + \sum_{j=1}^{l} x_{ij}(t_s + \Delta - h_i),\ i = 1, ..., k. \tag{3.47}$$

The term $\sum_{j=1}^{l} x_{ij}(t_s + \Delta - h_i)$ denotes UAVs that were send early, and that should come back due to their flight endurance. Otherwise, the term $x_{ij}(t_s + \Delta - h_i)$ means that $i - th$ UAV has sufficient endurance to continue service of $j - th$ zone, and hence, it can not come back to aerobase. Here we consider those objects where argument $t_s + \Delta - h_i > 0$. The initial conditions are $a_i(0) = a_i,\ i = 1, 2, ..., k$.

2) The number of UAVs that will serve the $j - th$ zone at the next moment $t_s + \Delta$ is composed of UAVs that are serving this zone at the previous moment t_s and having sufficient flight endurance, plus UAVs that reach this zone during the period $(t_s, t_s + \Delta]$, and minus UAVs that are out-of-fuel to the moment t. These facts lead the following equalities

$$b_j(t_s + \Delta) = b_j(t_s) - \sum_{i=1}^{k} x_{ij}(t_s - h_i + t_{ij}) + \sum_{i=1}^{k} x_{ij}(t_s - t_{ij}),\ j = 1, ..., l. \tag{3.48}$$

The term $\sum_{i=1}^{k} x_{ij}(t_s - h_i + t_{ij})$ denotes UAVs that should leave the $j - th$ zone due to their out-of-fuel. The term $\sum_{i=1}^{k} x_{ij}(t_s - t_{ij})$ denotes UAVs that were send early and should reach the $j - th$ zone during the period $(t_s, t_s + \Delta]$. Here we consider those objects where arguments $t_s - h_i + t_{ij} > 0$ and $t_s - t_{ij} > 0$. The initial conditions are $b_j(0) = b_j,\ j = 1, 2, ..., l$.

3) The variables $x_{ij}(t_s)$ at each moment $t_s, s = 1, ..., \nu$ satisfy the following conditions

$$\begin{aligned} a_i(t_s) + \sum_{j=1}^{l} x_{ij}(t_s) &= a_i,\ i = 1, ..., k. \\ b_j(t_s) + \sum_{i=1}^{k} x_{ij}(t_s - t_{ij}) &= b_j,\ (t_s - t_{ij} > 0)\ j = 1, ..., l. \end{aligned} \tag{3.49}$$

The first equation images the fact that the being UAVs can be allocated among zones. The second equation means that at each moment the service request should be satisfied.

The given above maim body of the problem constraints can be completed by additional conditions (constraints) followed from description of the Task 5.

4) Let τ_j^{first} is the given earliest time of 1-st visit to j zone for each $j,\ 1 \le j \le l$. Then the constraints (3.47)—(3.49) can be supplemented by the following:

$$\sum_{i=1}^{k} x_{ij}(s_j^{first}) \neq 0,\ 1 \le j \le l \tag{3.50}$$

where s_j^{first} is the discrete moment from the set $s = 1, 2, ..., \nu$ satisfying the following conditions:

$$s_j^{first}\Delta \le \tau_j^{first} \le (s_j^{first} + 1)\Delta \qquad \text{for some} \ \ s_j^{first} \in \{1, 2, ..., \nu\}.$$

The inequality (3.50) means that there exist at least one aerobase such that their UAVs will start with 1-st service visit to j zone no later on the preassigned moment τ_j^{first}.

Remark. If the τ_j^{first} is treated as the moment before of which the service of j zone is prohibited, then the constraints (3.47)—(3.49) can be supplemented by the following:

$$x_{ij}(s\Delta) = 0,\ 1 \le j \le l,\ \forall\ s\Delta \le \tau_j^{first} \text{ and } \ \forall\ i = 1, 2, ..., k \tag{3.51}$$

5) Let τ_j^{latest} is the given latest time of 1-st visit to j zone for each $j,\ 1 \le j \le l$. Then the constraints (3.47)—(3.49) can be supplemented by the following:

$$\sum_{i=1}^{k} x_{ij}(s_j^{first}) \neq 0, j,\ 1 \le j \le l \tag{3.52}$$

where s_j^{first} is the discrete moment from the set $s = 1, 2, ..., \nu$ satisfying the following conditions:

$$s_j^{first}\Delta \le \tau_j^{first} \le (s_j^{first} + 1)\Delta \qquad \text{for some} \ \ s_j^{first} \in \{1, 2, ..., \nu\}$$

and such that

$$s_j^{first} + h_i \le \tau_j^{latest},\ \ 1 \le j \le l,\ \forall\ i \in I_j^{first} \tag{3.53}$$

where

$$I_j^{first} = \{i,\ 1 \le i \le k : x_{ij}(s_j^{first}) \neq 0\}.$$

The couple of inequalities (3.52) — (3.53) means that there exist at least one aerobase such that their UAV the 1-st visit to j zone will begin no later the pre-assigned earliest time τ_j^{first}, and the ending this 1-st visit to j zone will be no later the pre-assigned the latest time τ_j^{latest} .

6) Let τ_j^{last} is the given last time of visits to j zone for each $j,\ 1 \le j \le l$. Then the constraints (3.47)—(3.49) can be supplemented by the following:

$$x_{ij}(s_j^{last}) = 0, \ \ \text{for all} \ \ s_j^{last} \le t_s \le \nu,\ \ 1 \le i \le k \ \text{ and } \ 1 \le j \le l \tag{3.54}$$

where s_j^{last} is the discrete moment from the set $s = 1, 2, ..., \nu$ satisfying the following conditions:

$$(s_j^{last} - 1)\Delta \leq \tau_j^{last} \leq s_j^{last}\Delta \qquad \text{for some} \;\; s_j^{last} \in \{1, 2, ..., \nu\}.$$

The equalities (3.54) denotes that all visits to j zone after the preassigned moment τ_j^{last} are prohibited.

3.3.4 Types of objective function

The cost value function used for optimization problem can be determined as follows:

a) the total service time for multiple zones

$$J_1(x) = \sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{s=0}^{\nu} x_{ij}(t_s)(h_i - 2t_{ij}). \tag{3.55}$$

b) the total number of UAVs "circles"

$$J_2(x) = \sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{s=0}^{\nu} x_{ij}(t_s) \tag{3.56}$$

c) the total unobservable time for multiple zones

$$J_3(x) = \sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{s=0}^{\nu} x_{ij}(t_s)(H - h_i - 2t_{ij}) \tag{3.57}$$

d) time of the first visit in the worst-case zone

$$J_4(x) = \max_{1\leq j\leq k} t_j^{first}, \quad \text{where } t_j^{first} = \min_{1\leq i\leq k}\min_{1\leq s\leq\nu}\{t_s : x_{ij}(t_s) \neq 0.\} \tag{3.58}$$

Hence, the optimization problem can be equipped by any of the proposed cost functions. In addition, some combinations of these function with the proper weighting coefficients can be used as a new cost function.

Thus, the optimal schedule problem of UAVs for multiple zones can be formulated, for example in the case of maximization of the total service time for multiple zones, as the following special integer dynamical linear programming problem (we change t_s by $t_s = s\Delta,\ s = 1, 2, ..., \nu$): maximize the cost value function

$$J_1(x) = \sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{s=0}^{\nu} x_{ij}(s\Delta)(h_i - 2t_{ij}) \to \max_{x_{ij}(s\Delta)\in\mathbb{N}, s=0,1,...,\nu} \tag{3.59}$$

subject to

$$
\begin{aligned}
a_i(s\Delta+\Delta) &= a_i(s\Delta) - \sum_{j=1}^{l} x_{ij}(s\Delta) + \sum_{j=1}^{l} x_{ij}(s\Delta+\Delta-h_i),\ i=1,...,k.\\
b_j(s\Delta+\Delta) &= b_j(s\Delta) - \sum_{i=1}^{k} x_{ij}(s\Delta-h_i+t_{ij}) + \sum_{i=1}^{k} x_{ij}(s\Delta-t_{ij}),\ j=1,...,l.\\
a_i(s\Delta) &+ \sum_{j=1}^{l} x_{ij}(s\Delta) = a_i,\ i=1,...,k.\\
b_j(s\Delta) &+ \sum_{i=1}^{k} x_{ij}(s\Delta-t_{ij}) = b_j,\ j=1,...,l,\\
& s=1,2,...,\nu,
\end{aligned}
\tag{3.60}
$$

and

$$\sum_{i=1}^{k} x_{ij}(s_j^{first}) \neq 0,\ 1\le j\le l \tag{3.61}$$

$$s_j^{first}+h_i \le \tau_j^{latest},\quad 1\le j\le l,\ \forall\ i\in I_j^{first}, \tag{3.62}$$

$$x_{ij}(s_j^{last})=0,\quad \text{for all}\quad s_j^{last}\le t_s\le\nu,\quad 1\le i\le k\quad\text{and}\quad 1\le j\le l \tag{3.63}$$

where
s_j^{first} is the discrete moment from the set $s=1,2,...,\nu$ satisfying the conditions

$$s_j^{first}\Delta \le \tau_j^{first} \le (s_j^{first}+1)\Delta \qquad \text{for some}\quad s_j^{first}\in\{1,2,...,\nu\},$$

s_j^{last} is the discrete moment from the set $s=1,2,...,\nu$ satisfying the conditions:

$$(s_j^{last}-1)\Delta \le \tau_j^{last} \le s_j^{last}\Delta \qquad \text{for some}\quad s_j^{last}\in\{1,2,...,\nu\}$$

and

$$I_j^{first}=\{i,\ 1\le i\le k : x_{ij}(s_j^{first})\neq 0\}.$$

Again note that $\nu=\left[\frac{H}{\Delta}\right]$ means the integer part of the fraction $\frac{H}{\Delta}$.

In (3.60) we consider those terms and elements where arguments $s\Delta-h_i+t_{ij}>0$ and $s\Delta-t_{ij}>0$.

Other optimization problem with another cost function mentioned above can be formulated by similar manner.

Remark 2.

The proposed partition of the planing horizon $[0,H]$ with small step Δ yields an ability to produce optimal schedule for UAVs, in fact, in regime of real time. The realization of this idea demands the development of some fast numerical algorithms for solution of the special classes of linear programming problems. Some new approaches to accelerate the solution of general linear programming problem is discussed in the paper [?]

Remark 3. In order to take into account the other request followed from description of the Task 5, the proposed model can be reformulated with the corresponding cost function. For example, the request to organize the zone service with maximum intervals between visits can be presented by maximization of the total unobservable time

$$J_3(x) = \sum_{i=1}^{k} \sum_{j=1}^{l} \sum_{s=0}^{\nu} x_{ij}(t_s)(H - h_i - 2t_{ij}) \to \max_{x_{ij}(s\Delta)\in\mathbb{N}, s=0,1,\ldots,\nu} \tag{3.64}$$

subject to constraints of (3.60) and (3.61).

The request to organize the zone service with minimum duration per visits can be presented by minimization of the total service time

$$J_1(x) = \sum_{i=1}^{k} \sum_{j=1}^{l} \sum_{s=0}^{\nu} x_{ij}(t_s)(h_i - 2t_{ij}) \to \min_{x_{ij}(s\Delta)\in\mathbb{N}, s=0,1,\ldots,\nu}. \tag{3.65}$$

subject to constraints of (3.60)and (3.61).

Remark 4. Another way to satisfy the multiple requests for zone service can be realized by optimization of one of the selected cost function and including the remained cost function into the main body of constraints (3.60)and (3.61) as follows, for example:

minimize the total number of UAVs "circles"

$$J_2(x) = \sum_{i=1}^{k} \sum_{j=1}^{l} \sum_{s=0}^{\nu} x_{ij}(t_s) \to \min_{x_{ij}(s\Delta)\in\mathbb{N}, s=0,1,\ldots,\nu}. \tag{3.66}$$

subject to constraints of (3.60)and (3.61) and the following new constraints

$$\sum_{i=1}^{k} \sum_{j=1}^{l} \sum_{s=0}^{\nu} x_{ij}(s\Delta)(h_i - 2t_{ij}) \leq A, \tag{3.67}$$

$$\sum_{i=1}^{k} \sum_{j=1}^{l} \sum_{s=0}^{\nu} x_{ij}(t_s)(H - h_i - 2t_{ij}) \leq B \tag{3.68}$$

where A and B are the known numbers those values can be given by specialists or can be determined experimentally.

Solution result representation

The obtained solution of optimization problem (3.59)—(3.61) can be presented in the form convenient for the practical uses. Such presentation can be done, for example, by Diagram or Schedule Table of time and duration visits by each MAS for the chosen zones.

Let $x^0_{ij}(t_s),\ s = 1, ..., \nu,\ i = 1, ..., k,\ j = 1, ..., l$ is optimal solution of (3.59)— (3.61) where $t_s = s\Delta$ and Δ is the sampling (discretisation) step.

First, we indicate resulting information concerning history of zone observation due to the obtained solution. For each zone j, where $1 \leq j \leq l$, define the following characteristics:

- $I_j = \{i \in \{1, 2, ..., k\} : x^0_{ij}(t_s) \neq 0, s = 1, 2..., \nu\}$ — the indexes of aerobase used for observation of j zone;
- $k_j = |I_j|$—the total number of aerobases involved in observation of j zone. Here $|I_j|$ denotes the number of elements of the set I_j);
- $m_j = |S_j|$, where $S_j = \{s \in \{1, 2, ..., \nu\} : x^0_{ij}(t_s) \neq 0, i = 1, 2..., k\}$ — the number of the discrete intervals of the form $[t_s, t_s + \Delta]$ during of which the observation by UAVs is realized for j zone. (Here $|S_j|$ denotes the number of elements of the set S_j);
- $T^{zone}_j = m_j\Delta$— the total duration of observation time for the j zone;
- $N^{zone}_j = \sum\limits_{i=1}^{k}\sum\limits_{s=1}^{\nu} x^0_{ij}(t_s)$—total number of UAVs used for observation of j zone;
- $t^{first}_j = \min\limits_{1\leq s\leq\nu}\{t_s : x^0_{ij}(t_s) \neq 0, i = 1, 2..., k\}$—the time of first visit to j zone;
- $t^{last}_j = \max\limits_{1\leq s\leq\nu}\{t_s : x^0_{ij}(t_s) \neq 0, i = 1, 2..., k\} + \Delta$—the time of the ending of observation of j zone;
- i^{first}_j —- those aerobases the UAVs of which were the first visitors of j zone, where i^{first}_j is the indexes from the set $\{1, 2, ..., k\}$ where the minimum for t^{first}_j is reached ;
- i^{last}_j — those aerobases the UAVs of which were the last visitors of j zone, where i^{last}_j is the indexes from the set $\{1, 2, ..., k\}$ where the maximum for the value t^{last}_j is reached.

It is possible, also, to restore more detail information concerning the visits of j zone between the first moment t^{first}_j of visit and the last moment t^{last}_j with detailed history for each observation intervals and UAVs involved in this observation.

On other hand, for UAVs addressed to observation mission can be useful information concerning their visits schedule for all zones for observation of which they are used.

For each i aerobase, where $1 \leq i \leq k$, define the following characteristics:

- $J_i = \{j \in \{1, 2, ..., l\} : x^0_{ij}(t_s) \neq 0, s = 1, 2..., \nu\}$ — the indexes of zone for observation of which the i aerobase is used;

$M_i = \sum_{i=1}^{k} \sum_{s=1}^{\nu} x_{ij}^0(t_s)$—total number of UAVs used for observation of j zone;

- $\tau_i^{first} = \min_{1 \le s \le \nu} \{t_s : x_{ij}^0(t_s) \ne 0, j = 1, 2..., l\}$—the time of the first mission fly of UAVs of i aerobase;
- j_i^{first} —- those zones, for which the UAVs of i aerobase are used firstly for observation mission, where j_i^{first} is the indexes from the set $\{1, 2, ..., l\}$ where the minimum for τ_i^{first} is reached ;

This list of characteristics for UAVs of i aerobase can be continued by obviously manner.

3.4 A case with single UAVs at aerobases

To simplify at this stage our calculations we suppose that every aerobase has one UAV. Otherwise, the aerobases where there are several UAVs can be formally divided onto the collection of several aerobases with alone UAV at every one. In the next chapter we will consider the general case, too.

3.4.1 Notation

Introduce the following notations:
n— number of aerobases,
K — number of zones for service,
V_k— number of UAVs which are required for service of k-th zone, $k = 1, ..., K$
$[\underline{T}_k, \overline{T}_k]$— "time window" for k-th zone where $\underline{T}_k$ and $\overline{T}_k$ is the earliest and latest time for service of k-th zone),
r_{jk}— distance from j-th aerobase to k-th zone,
d_{ij}— distance from i-th zone to j-th zone.

Introduce the network of aerobases and zones as a pair (S, U). Here $S = \{1, 2, ..., n, n+1, ..., n + K\}$- the set of numbered nodes- aerobases and zones, such that to each node corresponds aerobase or zone.
U-set of edges, which are connect the pair of nodes. The set S can be divided onto two subsets: S_A (set of aerobases) and S_Z (set of zones). Each node pair $(i, j), i \in S, j \in S$ corresponds the edge U_{ij} connecting the node i and node j. The edge U_{ij} have the characteristic ρ_{ij} — the distance between node i and j, i.e.
if $i \in S_A$ and $j \in S_Z$ then $\rho_{ij} = r_{ij}$;

if $i \in S_Z$ and $j \in S_Z$ then $\rho_{ij} = d_{ij}$.
Denote by
$\alpha_s, (s = 1, ..., n)$ — boolean variable where $\alpha_s = 1$ means that the $s-$ th aerobase (their UAV) involve into asked service, and $\alpha_s = 0$ — otherwise.
$\eta_i^{(s)}, (s = 1, ..., n;\ i = 1, ..., K$ — boolean variable where $\eta_i^{(s)} = 1$ means that the $s-$ th aerobase (their UAV) involve into service of $i-$ th zone, and $\eta_i^{(s)} = 0$ — otherwise.

3.4.2 Cost functions

Obviously, each assignment plan $\eta^{(s)} = \left(\eta_1^{(s)}, \eta_2^{(s)}, ..., \eta_K^{(s)}\right)$, $s = 1, ..., n$ of UAVs generates the boolean values α_s as follows

$$\alpha_s = \begin{cases} 1, \ if \ z^s > 0 \\ 0, \ if \ z^s = 0, \quad (s = 1, ..., n) \end{cases} \tag{3.69}$$

where $z^s = \sum\limits_{k=1}^{K} \eta_i^{(s)}$.

Then we can consider the cost functions

$$C_1(\eta) = \sum_{s=1}^{n} \alpha_s \tag{3.70}$$

that denotes the total number of UAVs used for service requests.

Next we introduce some other cost functions where it will be determined:
i) how many times each UAV is used in service
ii) total time service subject to constraints in the form of "time windows" for zone service.

To this aim we need to analyze some details of assignment plans in details.

3.4.3 Service logic and Constraints

Let $\eta^{(s)} = \left(\eta_1^{(s)}, \eta_2^{(s)}, ..., \eta_K^{(s)}\right)$ be an assignment plan for $s - th$ UAV (aerobase). Note, that the total number of all assignment plans for every aerobase is equal $K!$ (the number of all permutation of K elements). The value of $K!$ can be huge. By this reason, we can suppose that for each aerobase there exists some service order for considered zones. For example, this order can be determined in accordance with order of the assigned zone "time windows" such that the first for service is the zone with the smallest beginning of " window time". Some other ideas can be put to fix this order, also.

Next consider the time diagram of the considered flying route $\eta^{(s)}$.

Since in the considered route the zone-node $\eta_1^{(s)}$ is the first, and for this zone we have the time-window for service as $[\underline{T}_{\eta_1^{(s)}}, \overline{T}_{\eta_1^{(s)}}]$, then the time of the first departure from s-th base is:

$$t_1^{(s)} = \underline{T}_{\eta_1^{(s)}} - t_{fly}^{s \to \eta_1^{(s)}} \tag{3.71}$$

where $t_{fly}^{s \to \eta_1^{(s)}} = \dfrac{\rho_{s\eta_1^{(s)}}}{v_s}$ denotes the flying time from $s - th$ base to zone $\eta_1^{(s)}$.

Also it should be noted that it is not possible to start service of zone $\eta_1^{(s)}$ at the moment $\underline{T}_{\eta_1^{(s)}}$ if $t_1^{(s)} < 0$. But it is possible partially service if $h_s > t_{fly}^{s \to \eta_1^{(s)}} + t_{fly}^{\eta_1^{(s)} \to s}$, where h_s means the endurance of UAVs located at s-th base. If $t_1^{(s)} > 0$, then the service time of the first zone $\eta_1^{(s)}$ in the considered route $\eta^{(s)}$ is equal

$$T_{service}^{\eta_1^{(s)}} = \begin{cases} 0, \ if \ t_1^{(s)} < 0 \\ \overline{T}_{\eta_1}^{(s)} - \underline{T}_{\eta_1^{(s)}}, if \ t_1^{(s)} > 0 \ \text{ and } \ h_s > 2t_{fly}^{s \leftrightarrow \eta_1^{(s)}} + (\overline{T_{\eta_1}^{(s)}} - \underline{T_{\eta_1}^{(s)}}) \\ h_s - 2t_{fly}^{s \leftrightarrow \eta_1^{(s)}}, if \ t_0^{(s)} > 0 \quad \text{and} \ \ h_s < 2t_{fly}^{s \leftrightarrow \eta_1^{(s)}} + (\overline{T_{\eta_1}^{(s)}} - \underline{T_{\eta_1}^{(s)}}) \end{cases} \tag{3.72}$$

Thus, after analysis of the first node $\eta_1^{(s)}$ we can define the time of ending service for the first zone by s-th UAVs located at s-th base as follows:

$$t_{1,final}^{(s)} = \begin{cases} 0, \begin{cases} a) if \ t_1^{(s)} < 0 \text{(i.e. UAVs was not used for service of the first node)} \\ b) if \ t_1^{(s)} > 0 \ and \ h_s < 2t_{fly}^{s \leftrightarrow \eta_1^{(s)}} + (\overline{T_{\eta_1}^{(s)}} - \underline{T_{\eta_1}^{(s)}}) \\ \text{(i.e. UAVs was used at first zone and then it returned to base} \\ \text{ due to restricted endurance)} \end{cases} \\ (t_{fly}^{s \to \eta_1^{(s)}} + T_{service}^{s \to \eta_1^{(s)}}), \text{(i.e. when endurance of UAV was more then required for} \\ \text{service zone } \eta_1^{(s)} \text{ and UAV can fly for service from zone } \eta_1^{(s)} \text{ to next zone } \eta_2^{(s)}) \end{cases} \tag{3.73}$$

Now consider how we can to start the service of the next zone from our route $\eta^{(s)}$ taking into account the previous analysis and (3.73). Find the starting moment

$$t_{start}^{\eta_2^{(s)}} = \begin{cases} \underline{T_{\eta_2}^{(s)}} - t_{fly}^{s \to \eta_2^{(s)}}, \text{ if } \ t_{1,final}^{(s)} = 0 \text{(i.e. this is the case,} \\ \text{when we are "start" from the base)} \\ t_{fly}^{s \to \eta_1^{(s)}} + T_{service}^{\eta_1^{(s)}}, \ \text{ otherwise (namely we are starting from the first zone } \eta_1^{(s)}) \end{cases} \tag{3.74}$$

It should be noted once again that, if $t_{start}^{\eta_2^{(s)}} < 0$, then this zone will be eliminated from further consideration, since the considered $s - th$ UAV does not reach this zone. In the case when $t_{start}^{\eta_2^{(s)}} > 0$ we can to continue the analysis of possibilities of servicing node (zone) $\eta_2^{(s)}$ taking into account the "time window" constraint $[\underline{T_{\eta_2}^{(s)}}, \overline{T_{\eta_2}^{(s)}}]$.

Then

$$T_{service}^{\eta_2^{(s)}} = \begin{cases} a) 0, if\ t_{start}^{\eta_2^{(s)}} < 0 \\ b) \overline{T_{\eta_2}^{(s)}} - \underline{T_{\eta_2}^{(s)}}, if\ t_{start}^{\eta_2^{(s)}} > 0\ and\ t_{1,final}^{(s)} = 0\ and\ h_s > 2t_{fly}^{s \leftrightarrow \eta_2^{(s)}} + (\overline{T_{\eta_2}^{(s)}} - \underline{T_{\eta_2}^{(s)}}) \\ \text{i.e the case, when we will start from the base} \\ \text{and we have sufficient endurance to serve the node } \eta_2^{(s)} \text{ and coming back to base} \\ c) h_s - 2t_{fly}^{s \to \eta_1^{(s)}}, if\ t_{start}^{\eta_2^{(s)}} > 0\ and\ t_{1,final}^{(s)} = 0\ but\ h_s < 2t_{fly}^{s \to \eta_2^{(s)}} + (\overline{T_{\eta_2}^{(s)}} - \underline{T_{\eta_2}^{(s)}}) \\ \text{(i.e. the case when not completely "close" the window....)} \\ d) h_s - t_{fly}^{s \to \eta_1^{(s)}} - t_{fly}^{\eta_1 \to \eta_2} - t_{fly}^{\eta_2 \to s}, \text{if served } \eta_2 \text{ from } \eta_1 \\ \text{ and then back to base } s, \\ \text{ since there was not sufficient endurance to continue service} \\ e) T_{service}^{\eta_1^{(s)}} + (\overline{T_{\eta_2}^{(s)}} - \underline{T_{\eta_2}^{(s)}}), \text{if served the node } \eta_2 \text{ from } \eta_1 \text{ and have sufficient endurance.} \end{cases} \quad (3.75)$$

Continue by analogy with above the given analysis for the remainder zones from the considered route $\eta^{(s)}$ we find the sequence

$$T_{service}^{\eta_1^{(s)}}, T_{service}^{\eta_2^{(s)}}, ..., T_{service}^{\eta_K^{(s)}}$$

of time services of each zones. Then the total service time which generates the considered route $\eta^{(s)}$ is

$$T_{service}(\eta^{(s)}) = T_{service}^{\eta_1^{(s)}} + T_{service}^{\eta_2^{(s)}} + ... + T_{service}^{\eta_K^{(s)}} \quad (3.76)$$

and, hence, the total service time of the required zones is

$$T_{service} = \sum_{s=1}^{n} T_{service}(\eta^{(s)}) \quad (3.77)$$

To guarantee the needed number V_k of pre-assigned UAVs for $k - th$ zone we should set the following constraints for the introduced boolean variables

$$\sum_{s=1}^{n} \eta_k^{(s)} = V_k,\ k = 1.2, ..., K \quad (3.78)$$

Finally, the assignment problem with timing constraints can be formulated as the following boolean optimization problem; Maximize the total service time

$$\sum_{s=1}^{n} T_{service}(\eta^{(s)}) \rightarrow \max_{\eta^{(s)} \in \text{boolean}} \tag{3.79}$$

subject to constraints

$$\sum_{s=1}^{n} \eta_k^{(s)} = V_k, \;\; k = 1.2, ..., K \tag{3.80}$$

Remark 1. *The problem (3.79)-(3.80) is generalized case of the static problem from D5 report, namely:*

To find $x_{ij}, (i = 1, 2, ..., k; j = 1, 2, ..., l)$ such that, the total cost function for all services performed by all UAVs takes an optimal value

$$F = \sum_{i=1}^{k} \sum_{j=1}^{l} c_{ij} x_{ij} \rightarrow \min_{x_{ij}} \tag{3.81}$$

subject to

$$\sum_{i=1}^{k} x_{ij} = b_j, \quad j = 1, 2, ..., l \tag{3.82}$$

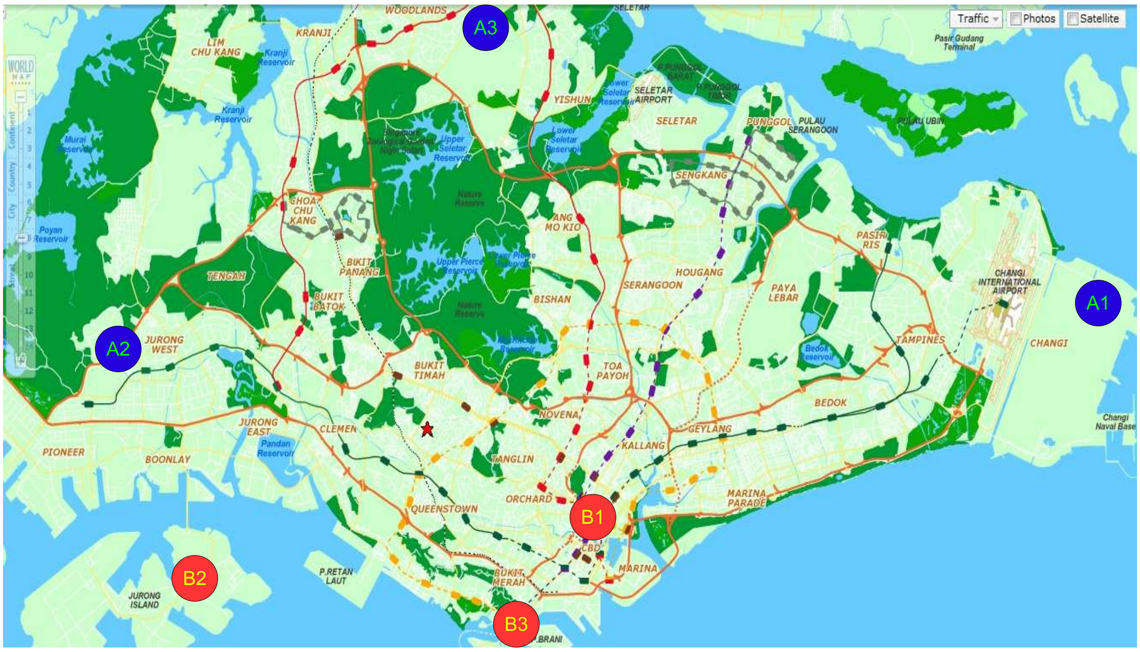
$$\sum_{j=1}^{l} x_{ij} = a_i, \quad i = 1, 2, ..., k$$

$$\sum_{i=1}^{k} a_i = \sum_{j=1}^{l} b_j$$

$$x_{ij} \geq 0, \qquad x_{ij} \;\; \textit{are integer numbers.}$$

3.5 Illustrative examples

Assume that we have 3 airbases located at Changi A_1 with 3 UAVs ($a_1 = 3$), Jurong West A_2 with 3 UAVs ($a_2 = 3$), and Woodland A_3 with 1 UAV ($a_1 = 1$). Now 7 UAVs are requested from B_1-Raffles Place ($b_1 = 2$), B_2-Jurong Island ($b_2 = 2$), and B_3- Sentosa Island ($b_3 = 3$). Out task is to complete all requests in order to maximaze the total service time in the zones and satisfies all timing constraints.



Also we assume that initial data are the same as in the previous case. Namely, the distances between A_i and B_j are given as follows (in kilometers):

	A_1	A_2	A_3
A_1	0	32	22
A_2	32	0	17
A_3	22	17	0

Distances between A_i

	B_1	B_2	B_3
B_1	0	17	6
B_2	17	0	14
B_3	6	14	0

Distances between B_j

	B_1	B_2	B_3
A_1	13	30	18
A_2	16	9	17
A_3	21	20	23

Distances between A_i and B_j

The speed of UAVs is fixed $v_{ij} = 30\frac{m}{sec}$.

Next, for all i and j denote by c_{ij} the benefit of sending the UAV from i-th aerobase to j-th zone of area of operation. In particular, this benefit can be given in the form $c_{ij} = \frac{d_{ij}}{v_{ij}}$ that means the flight time from $A_i \to B_j$ (see Table):

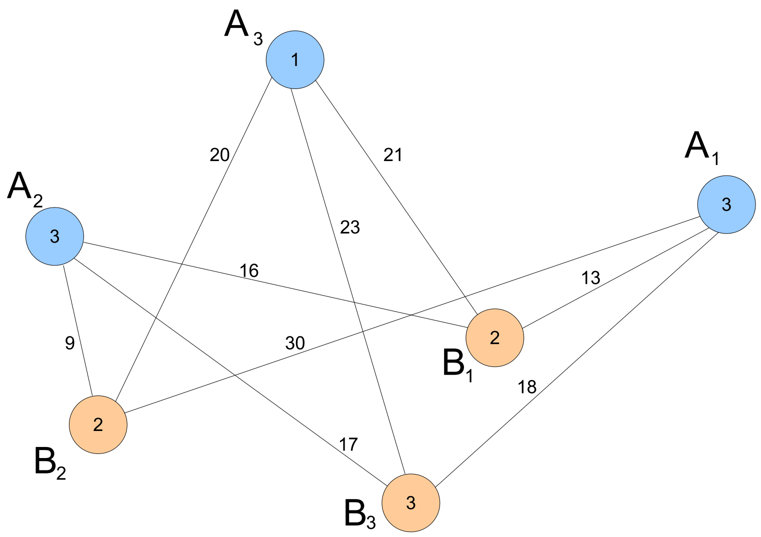
	B_1	B_2	B_3
A_1	433	1000	600
A_2	533	300	566
A_3	700	666	766

(3.83)

UAVs flight time from $A_i \to B_j$

The optimization problem is to optimize the total service time of the group of UAVs.

It should be noted that the total service time of the group of UAVs involved in the mission will be optimal if the total flight time used to reach the preassigned zones is minimal.

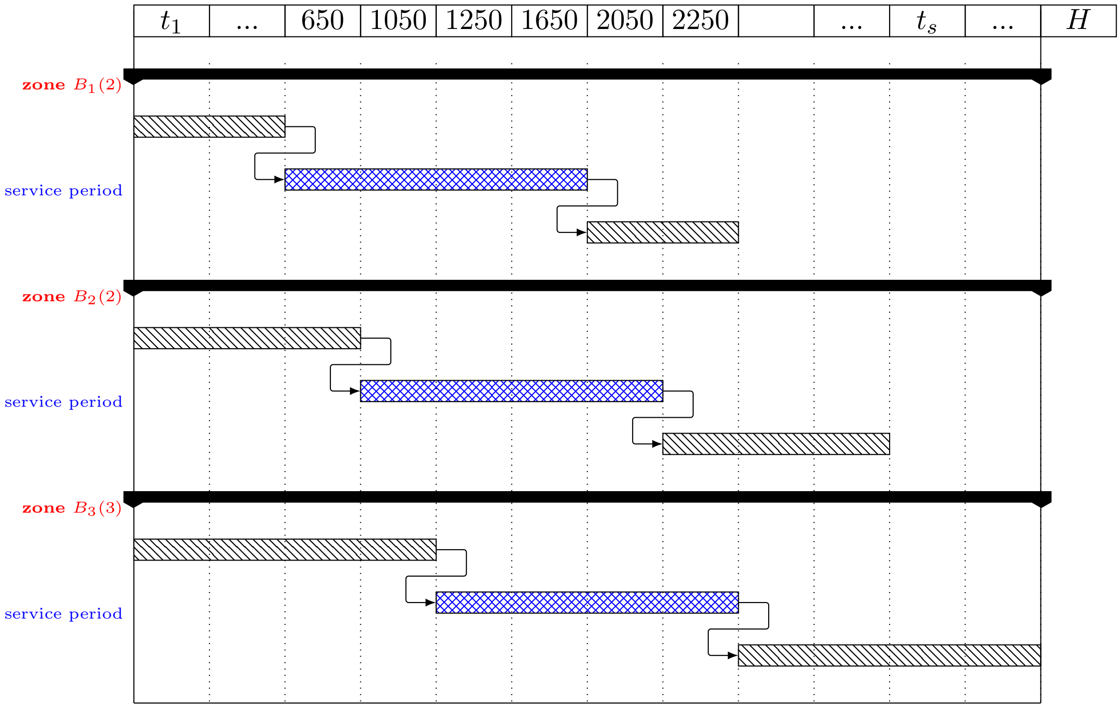


We will use the following notation for this problem:

$A_i, i = 1, 2, 3$- number of aerobases,
$a_1 = 3, a_2 = 3, a_3 = 1$ - number of UAVs located in A_i,
$B_j, j = 1, 2, 3$- areas of operations,
$b_1 = 2, b_2 = 2, a_3 = 3$- numbers of UAVs for service of B_j
d_{ij}- distances from A_i to B_j given in table 3;
x_{ij}-number of UAVs from A_i to B_j
c_{ij}- given in table 4. $h_i = 3600sec$- UAVs endurance located on A_i aerobase;
$v_{ij} = 30\frac{m}{sec}$ - speed of UAVs;
$t^f_{B_i}$- earliest time for visit zone $B_i, i = 1, 2, 3$;
$t^l_{B_i}$- latest time for visit zone $B_i, i = 1, 2, 3$
Let the following value for "time windows":

$$\begin{aligned} t^f_{B_1} &= 650sec, \quad t^l_{B_1} = 1650sec; \\ t^f_{B_2} &= 1050sec, \quad t^l_{B_2} = 2050sec; \\ t^f_{B_3} &= 1250sec, \quad t^l_{B_3} = 2250sec; \end{aligned} \tag{3.84}$$

This "time windows" requirements are shown on the Diagram:



From this diagram it is clear that we can divide our problem by considering the assignments problem on the following 5 periods:

Period 1: [650,1050] - 1 problem for zone B_1 to assign 2 UAVs (i.e. $B_1(2)$);

Period 2: [1050,1250] - 2 problems for zones B_1 and B_2 (to assign 4 UAVs – $B_1(2)$, $B_2(2)$);

Period 3: [1250,1650] - 3 problems for zones $B_1(2)$, $B_2(2)$ and $B_3(3)$;

Period 4: [1650,2050] - 2 problems for zones $B_2(2)$ and $B_3(3)$;

Period 5: [2050,2250] - 1 problem for zone $B_3(3)$.

Applying for each period the NSW method together with method of potentials described in D5 report we have the following schedular plan:

$A_1(3) \xrightarrow{2} B_1(2)$,

$A_1(3) \xrightarrow{1} B_3(3)$,

$A_2(3) \xrightarrow{2} B_2(2)$,

$A_2(3) \xrightarrow{1} B_3(3)$,

$A_3(1) \xrightarrow{1} B_3(3)$.

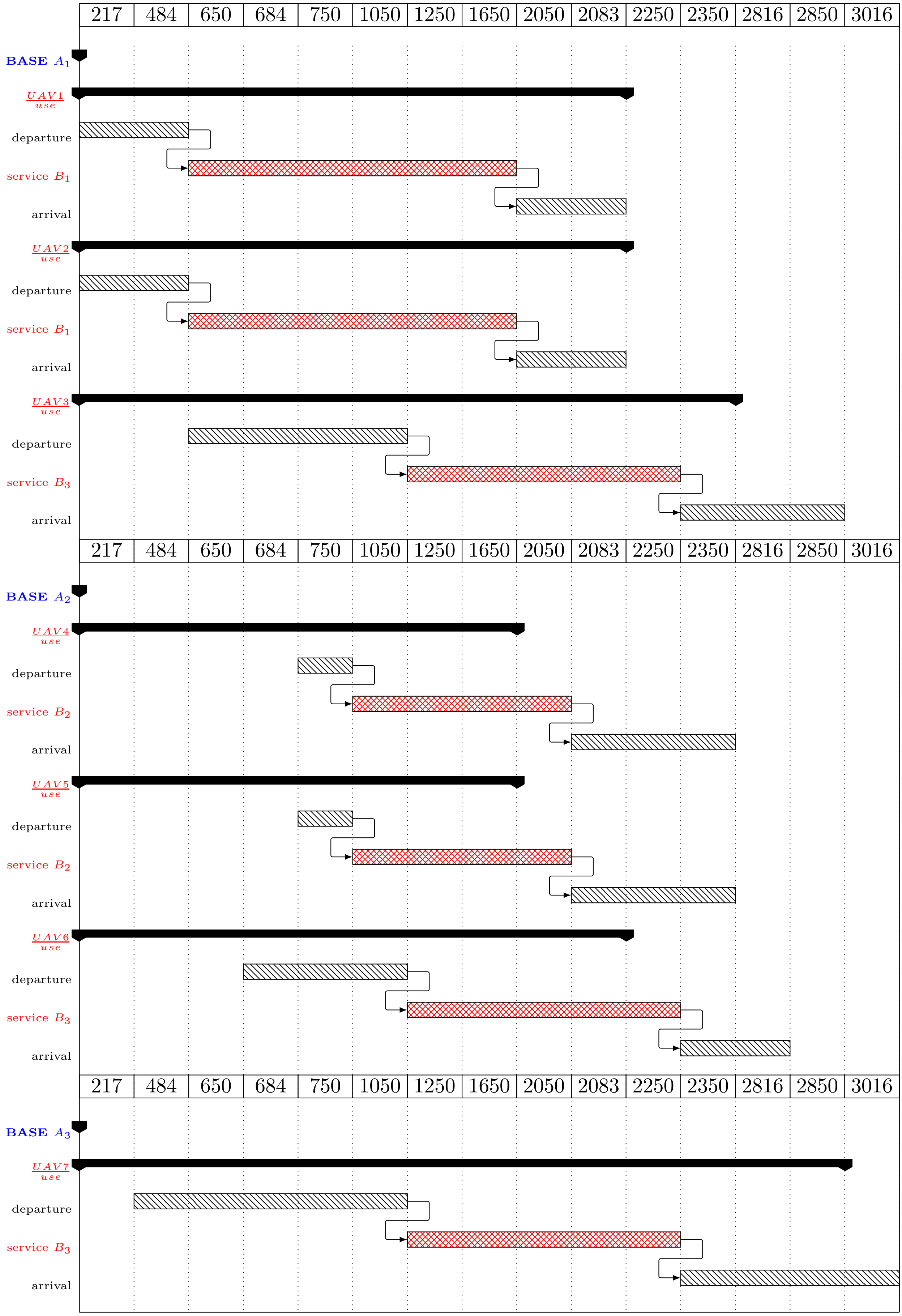
This schedular plan can be written in the following Table:

	B_1	B_2	B_3	a_i
A_1	2	0	1	$a_1 = 3$
A_2	0	2	1	$a_2 = 3$
A_3	0	0	1	$a_3 = 1$
b_j	$b_1 = 2$	$b_2 = 2$	$b_3 = 3$	$\sum_{i=1}^{3} a_i = \sum_{j=1}^{3} b_j = 7$

To observe the time schedular for each UAV, this plan can be written in the Table of the form:

		B_1		B_2		B_3	
		Departure time (D/T)	Arrival time (A/T)	D/T	A/T	D/T	A/T
A_1	UAV 1	217	2083	-	-	-	-
	UAV 2	217	2083	-	-	-	-
	UAV 3	-	-	-	-	650	2850
A_2	UAV 4	-	-	750	2350	-	-
	UAV 5	-	-	750	2350	-	-
	UAV 6	-	-	-	-	684	2816
A_3	UAV 7	-	-	-	-	484	3016

Also, this schedular can be imaged by the following Diagram where the exact flying plan is given for all of UAVs:


217
484
650
684
750
1050
1250
1650
2050
2083
2250
2350
2816
2850
3016
BASE A_1
$\frac{UAV1}{use}$
departure
service B_1
arrival
$\frac{UAV2}{use}$
departure
service B_1
arrival
$\frac{UAV3}{use}$
departure
service B_3
arrival
BASE A_2
$\frac{UAV4}{use}$
departure
service B_2
arrival
$\frac{UAV5}{use}$
departure
service B_2
arrival
$\frac{UAV6}{use}$
departure
service B_3
arrival
BASE A_3
$\frac{UAV7}{use}$
departure
service B_3
arrival

The total service time performed by all UAVs takes an optimal value

$$
\begin{aligned}
T^{service} = \sum_{i=1}^{7} h_i - 2 \min_{x_{ij}} \sum_{i=1}^{3} \sum_{j=1}^{3} \frac{d_{ij}}{v_{ij}} x_{ij} - \sum_{i=1}^{7} T_i^{zone} = \\
= 7 * 3600 - 2 * 3398 - 7 * 1000 \; sec. \\
\approx 3,18 \; hours
\end{aligned}
\tag{3.85}
$$

Chapter 4

Reducing the dynamical optimization problem to static optimization problem

The proposed dynamical transportation problem (3.59)—(3.61) for allocation of MAS can be presented as a static problem given in the previous D5 report. But this way leads to the huge dimensions of the variables involved, and this together the specific structure of the considered problem are a serious obstacle for suitable solution for reasonable time. By this reason the development of special methods and design on this base of fast numerical methods for assignment problems of MAS with next their realization in the corresponding computer chips are actual and will be done at this work.

4.1 Matrix form

Introduce the following matrixes

$$A_{k\times\nu} = \begin{pmatrix} a_1(\Delta) & a_1(2\Delta) & ... & a_1(\nu\Delta) \\ a_2(\Delta) & a_2(2\Delta) & ... & a_2(\nu\Delta) \\ a_3(\Delta) & a_3(2\Delta) & ... & a_3(\nu\Delta) \\ ... & ... & ... & ... \\ a_k(\Delta) & a_k(2\Delta) & ... & a_k(\nu\Delta) \end{pmatrix}, \tag{4.1}$$

$$H^{-}_{\nu\times(\nu-1)} = \begin{pmatrix} 0 & 0 & 0 & ... & 0 \\ 1 & 0 & 0 & ... & 0 \\ 0 & 1 & 0 & ... & 0 \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & 1 \end{pmatrix}, \tag{4.2}$$

$$H^{+}_{\nu\times(\nu-1)} = \begin{pmatrix} 1 & 0 & 0 & ... & 0 \\ 0 & 1 & 0 & ... & 0 \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & 1 \\ 0 & 0 & 0 & ... & 0 \end{pmatrix}, \tag{4.3}$$

$$X_i = \begin{pmatrix} x_{i1}(\Delta) & x_{i1}(2\Delta) & ... & x_{i1}(\nu\Delta) \\ x_{i2}(\Delta) & x_{i2}(2\Delta) & ... & x_{i2}(\nu\Delta) \\ ... & ... & ... & ... \\ x_{il}(\Delta) & x_{il}(2\Delta) & ... & x_{il}(\nu\Delta) \end{pmatrix}_{l\times\nu}, i = 1, ..., k \tag{4.4}$$

Introduce the block matrixes of the form

$$X = \begin{pmatrix} X_1 \\ X_2 \\ ... \\ X_k \end{pmatrix}_{kl\times\nu}, \ \Pi = \begin{pmatrix} e_l & 0_l & ... & 0_l \\ 0_l & e_l & ... & 0_l \\ ... & ... & ... & ... \\ 0_l & 0_l & ... & e_l \end{pmatrix}_{k\times kl} \tag{4.5}$$

where

$$e_l = \begin{pmatrix} 1 & 1 & ... & 1 \end{pmatrix}_{1\times l}, 0_l = \begin{pmatrix} 0 & 0 & ... & 0 \end{pmatrix}_{1\times l}, \tag{4.6}$$

Remark.

Since the unknown variables of the optimization problem are $x_{ij}(\Delta), x_{ij}(2\Delta), ..., x_{ij}(\nu\Delta)$, then the other variables $x_{ij}(t)$ with argument t that is not coincide with arguments $\Delta, 2\Delta, ..., \nu\Delta$ will be approximated by the variables $x_{ij}(s\Delta)$ where $s = \left[\dfrac{t}{\Delta}\right]$ is the integer part of the number $s = \dfrac{t}{\Delta}$ such that the argument $s\Delta$ is the nearest to the argument

t. Such kind approximation is admissible due to the freedom in choice of sampling step Δ. We assume, in fact, that for the considered optimization problem the unknown continuous function $x_{ij}(\tau)$ of the real variable τ can be approximated by piecewise constant function $x_{ij}(s\Delta),\ s = 1, ..., \nu$.

Noting the given remark, introduce the following matrixes

$$
\begin{aligned}
h(X_i) &= \begin{pmatrix}
x_{i1}\left(\Delta\left[\dfrac{\Delta - h_i}{\Delta}\right]\right) & x_{i1}\left(\Delta\left[\dfrac{2\Delta - h_i}{\Delta}\right]\right) & ... & x_{i1}\left(\Delta\left[\dfrac{\nu\Delta - h_i}{\Delta}\right]\right) \\
x_{i2}\left(\Delta\left[\dfrac{\Delta - h_i}{\Delta}\right]\right) & x_{i2}\left(\Delta\left[\dfrac{2\Delta - h_i}{\Delta}\right]\right) & ... & x_{i2}\left(\Delta\left[\dfrac{\nu\Delta - h_i}{\Delta}\right]\right) \\
... & ... & ... & ... \\
x_{il}\left(\Delta\left[\dfrac{\Delta - h_i}{\Delta}\right]\right) & x_{il}\left(\Delta\left[\dfrac{2\Delta - h_i}{\Delta}\right]\right) & ... & x_{il}\left(\Delta\left[\dfrac{\nu\Delta - h_i}{\Delta}\right]\right)
\end{pmatrix}_{l\times\nu}, \\
i &= 1, ..., k
\end{aligned}
$$

Introduce the block matrixes of the form

$$
h(X) = \begin{pmatrix} h(X_1)H^- \\ h(X_2)H^- \\ ... \\ h(X_k)H^- \end{pmatrix}_{kl\times(\nu)}, \ a_{k\times 1} = \begin{pmatrix} a_1 \\ a_2 \\ ... \\ a_k \end{pmatrix}, e_{1\times\nu}\begin{pmatrix} 1 & 1 & ... & 1 \end{pmatrix} \tag{4.7}
$$

Then the first and third equations of (3.60) can be written in the matrix form as follows

$$
AH^- = AH^+ - \Pi X + \Pi h(X) \tag{4.8}
$$

$$
A + \Pi X = a_{k\times 1}e_{1\times\nu} \tag{4.9}
$$

In order to rewrite the remained equations of (3.60) introduce the matrixes

$$
\begin{aligned}
B &= \begin{pmatrix} b_1(\Delta) & b_1(2\Delta) & ... & b_1(\nu\Delta) \\ b_2(\Delta) & b_2(2\Delta) & ... & b_2(\nu\Delta) \\ b_3(\Delta) & b_3(2\Delta) & ... & b_3(\nu\Delta) \\ ... & ... & ... & ... \\ b_l(\Delta) & b_l(2\Delta) & ... & b_l(\nu\Delta) \end{pmatrix}_{l\times\nu} \\
T(X_i) &= \begin{pmatrix} x_{i1}\Big(\Delta\Big[\dfrac{\Delta - t_{i1}}{\Delta}\Big]\Big) & x_{i1}\Big(\Delta\Big[\dfrac{2\Delta - t_{i1}}{\Delta}\Big]\Big) & ... & x_{i1}\Big(\Delta\Big[\dfrac{\nu\Delta - t_{i1}}{\Delta}\Big]\Big) \\ x_{i2}\Big(\Delta\Big[\dfrac{\Delta - t_{i1}}{\Delta}\Big]\Big) & x_{i2}\Big(\Delta\Big[\dfrac{2\Delta - t_{i1}}{\Delta}\Big]\Big) & ... & x_{i2}\Big(\Delta\Big[\dfrac{\nu\Delta - t_{i1}}{\Delta}\Big]\Big) \\ ... & ... & ... & ... \\ x_{il}\Big(\Delta\Big[\dfrac{\Delta - t_{i1}}{\Delta}\Big]\Big) & x_{il}\Big(\Delta\Big[\dfrac{2\Delta - t_{i1}}{\Delta}\Big]\Big) & ... & x_{il}\Big(\Delta\Big[\dfrac{\nu\Delta - t_{i1}}{\Delta}\Big]\Big) \end{pmatrix}_{l\times\nu} \\
TH(X_i) &= \begin{pmatrix} x_{i1}\Big(\Delta\Big[\dfrac{\Delta - h_i + t_{i1}}{\Delta}\Big]\Big) & x_{i1}\Big(\Delta\Big[\dfrac{2\Delta - h_i + t_{i1}}{\Delta}\Big]\Big) & ... & x_{i1}\Big(\Delta\Big[\dfrac{\nu\Delta - h_i + t_{i1}}{\Delta}\Big]\Big) \\ x_{i2}\Big(\Delta\Big[\dfrac{\Delta - h_i + t_{i2}}{\Delta}\Big]\Big) & x_{i2}\Big(\Delta\Big[\dfrac{2\Delta - h_i + t_{i2}}{\Delta}\Big]\Big) & ... & x_{i2}\Big(\Delta\Big[\dfrac{\nu\Delta - h_i + t_{i2}}{\Delta}\Big]\Big) \\ ... & ... & ... & ... \\ x_{il}\Big(\Delta\Big[\dfrac{\Delta - h_i + t_{il}}{\Delta}\Big]\Big) & x_{il}\Big(\Delta\Big[\dfrac{2\Delta - h_i + t_{il}}{\Delta}\Big]\Big) & ... & x_{il}\Big(\Delta\Big[\dfrac{\nu\Delta - h_i + t_{il}}{\Delta}\Big]\Big) \end{pmatrix} \\
i &= 1, ..., k
\end{aligned}
$$

and

$$
T(X) = \begin{pmatrix} T(X_1) \\ T(X_2) \\ ... \\ T(X_k) \end{pmatrix}_{lk\times\nu}, \; TH(X) = \begin{pmatrix} TH(X_1) \\ TH(X_2) \\ ... \\ TH(X_k) \end{pmatrix}_{lk\times\nu}
$$

Then the second an forth equations of (3.60) can be written as

$$
BH^- = BH^+ - \Pi TH(X) + \Pi T(X), \tag{4.10}
$$

$$
B + \Pi T(X) = b_{l\times 1} e_{1\times\nu} \tag{4.11}
$$

where

$$
b_{l\times 1} = \begin{pmatrix} b_1 \\ b_2 \\ ... \\ b_l \end{pmatrix}, e_{1\times\nu} \begin{pmatrix} 1 & 1 & ... & 1 \end{pmatrix}
$$

Finally, for example, the cost function $J_2(x) = \sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{s=1}^{\nu} x_{ij}(t_s)$ can be written as

$$J_2(X) = e_{kl}^T X e_{kl} \tag{4.12}$$

where $e_{kl}^T = \begin{pmatrix} 1 & 1 & ... & 1 \end{pmatrix}_{1\times kl}$ is the unit vector.

Thus, the matrix optimization problem is to find the integer valued matrix X maximizing the cost function

$$J_2(X) = e_{kl}^T X e_{kl} \to \max_X \tag{4.13}$$

subject to

$$\begin{aligned}
A(H^- - H^+) &= \Pi\left(h(X) - X\right) && (4.14)\\
A - a_{k\times 1}e_{1\times\nu} &= -\Pi X && (4.15)\\
B(H^- - H^+) &= \Pi\left(T(X) + TH(X)\right), && (4.16)\\
B - b_{l\times 1}e_{1\times\nu} &= -\Pi T(X) && (4.17)
\end{aligned}$$

It can be shown also that the problem above can be rewritten in the coordinate form and reduced to the following problem

$$\begin{aligned}
C^T X &\to \min, && (4.18)\\
AX &= B,\\
D_* \leq x &\leq D^*.
\end{aligned}$$

Remark 2. *The problem (4.18) can be solved more effectively by adaptive method (i.e. number of iteration, CPU time, etc.), which are presented in next chapter.*

Chapter 5

Comparison of the adaptive method with classical methods for linear programming

The main purpose of this chapter is to comparison of two methods for solving linear optimization problem (4.18). Namely, the adaptive method based on the constructive approach and well known classical simplex method in canonical form. It should be noted that adaptive method belongs to the same class as a primal simplex method (Danzig,1963). However, the author's of adaptive method are avoid the most popular verification of simplex method and call on to the well known principle in nonlinear programming - principle of admissible(feasible) direction. It is known that algorithm based on the principle of admissible direction can work on arbitrary feasible points, in contrast to the simplex method which based on special basis feasible points. Another significant difference of these method is that adaptive algorithm possesses a suboptimal criterion which stops the algorithm with the desired accuracy. From other hand, to stop the solution process the simplex method uses (in the case of existence of solution) only the optimality criteria since it has no suboptimality criteria at all.

The linear optimization methods among the modern optimization methods are most theoretically developed and practically implemented. The linear programming are connected to the optimization problem of linear function on a set given by linear equations and inequalities. It was designed in 40-50th last century. The linear programming models plays at those time an exceptionally important role in practical applications as a fundamental tool for maximizing resources and profit. The history and the ways of developing of the linear programming can be found in L.V. Kantorovich works who devoted his life to the struggle for recognition of new scientific methods of planning and organizing economy discovered by him and in monograph of another great mathematician - George B. Danzig, who known as a father of linear programming and inventor of the simplex method.

Initially, the idea of the method has been realized for the canonical problem of LP with one-sided constraints

$$c^T x \to \max, \qquad Ax = b, \qquad x \geq 0 \tag{5.1}$$
$$(b \geq 0, \quad A \in \mathbf{R}^{m\times n}, \quad rankA = m < n).$$

Afterwards, the simplex method has been extended for the canonical problem of LP with two-sided constraints

$$\begin{aligned} c^T x &\to \max, \\ Ax &= b, \\ d_* \leq x &\leq d^*. \end{aligned} \tag{5.2}$$

A comparison of the adaptive method and the more commonly used classical simplex algorithm for solving linear programming problems based under the natural assumption that for the solution of the practical problems used not only the mathematical problem statement but also the priory information of the feasible points. These condition can be treated as an experience of the functioning of the system, the knowledge of the specialists, also guess and intuition of the experts, the solution of the same problem in more simple form, etc. Actually, by this reason the method called adaptive since its properties of using the all the initial and current information for effective construction of suboptimal feasible solution.

5.1 Simplex method

Now briefly give the definition which are used in Simplex method.

Denote by $X \subset \mathbf{R}^n$ the set of the form

$$X = \{x \in \mathbf{R}^n : Ax = b, d_* \leq x \leq d^*\}. \tag{5.3}$$

The elements from the set X are called the feasible solution(points). The feasible points are satisfies both the general ($Ax = b$) and the simple ($d_* \leq x \leq d^*$) constraints.

Definition 1. *The feasible point x^o will be called optimal solution of the problem (5.2) if the objective function achieves the maximal value at this point.*

Denote by $I = (1, 2, ..., m)$, $J = (1, 2, ..., n)$ the corresponding set of indices for rows and columns of the matrix A. The constraint matrix A has m rows (constraints) and n columns (variables).

Definition 2. *$x = x(J)$ of $(Ax = b)$ is a basic solution if the n components of x can be partitioned into m "basic" and $n - m$ "non-basic" variables in such a way that:*

- *the $n - m$ components of x takes the limit value $x_j = d_{*j} \vee d^*_j, j \in J_N, |J_N| = n - m$. And*
- *to another components of x, namely to the $x_j, j \in J_B = J \setminus J_N, |J_B| = m$ corresponding the linear independent columns $a_j = A(I, j), j \in J_B$ of the matrix A.*

The indices $j \in J_B$ and components $x_j, j \in J_B$ of the basic solution x are called basis indices and components; $j, x_j, j \in J_N$ are called non-basis indices and components; the matrix $A_B = A(I, J_B)$ are called basis matrix; $A_N = A(I, J_N)$ are called non-basis matrix; the set J_B is called basis set; the set J_N is called non-basis set.

Definition 3. *The basis solution x is called non-degenerate, if its basis components are not critical $d_{*j} < x_j < d^*_j, j \in J_B$.*

The following steps are constructive. They illustrate how to generate a search direction Δx that is a *descent direction* (improving direction) for the objective $c^T x$. Primal Simplex does this by considering "Shall we move one of the nonbasic variables either up or down".

If there is no such direction, the current x is an optimal solution. Otherwise it is good to move as far as possible along the search direction, because $c^T x$ is linear. Usually a basic variable reaches a bound and a basis exchange takes place. The process then repeats.

Search direction Δx Consider along with feasible solution x another one $\bar{x} = x + \Delta x$. Then from $Ax = b, A\bar{x} = b$ follows that Δx satisfy :

$$A\Delta x = A(\bar{x} - x) = A\bar{x} - Ax = b - b = 0 \tag{5.4}$$

or in component form

$$A_B \Delta x_B + A_N \Delta x_N = 0, \quad \text{where} \quad x_B = x(J_B), x_N = x(J_N). \tag{5.5}$$

Since the $det A_B \neq 0$ the basis component can be presented as

$$\Delta x_B = -A_B^{-1} A_N \Delta x_N. \tag{5.6}$$

Effect on objective Insert last representation of Δx_B to the increment formula:

$$c^{'}\bar{x} - c^{'}x = c^{'}\Delta x = c^{'}_B \Delta x_B + c^{'}_N \Delta x_N = -(c^{'}_B A_B^{-1} A_N - c^{'}_N)\Delta x_N. \tag{5.7}$$

Introduce next the vector of potentials and vectors of estimates as:

$$u = u(I) : u^{'} = c^{'}_B A_B^{-1}; \tag{5.8}$$

$$\Delta = \Delta(J) : \Delta^{'} = u^{'} A_N - c^{'}_N \tag{5.9}$$

The increment formula (5.7) can be rewritten in more compact form:

$$c^{'}\Delta x = -\Delta_N \Delta x_N = -\sum_{j \in J_N} \Delta_j \Delta x_j. \tag{5.10}$$

Optimality Using (5.10) it is easy to proof the following theorem:

Theorem 1. *For the optimality of a basis feasible point x it is sufficient and, in the case of non-degeneracy of it, also necessary, that the following conditions:*

$$\begin{cases} \Delta_j \geq 0 \ \ \textit{for } x_j = d_{*j}, \\ \Delta_j \leq 0 \ \ \textit{for } x_j = d^*_j, j \in J_N \end{cases} \tag{5.11}$$

holds.

Another words, no improvement is possible if one of the above conditions holds for every nonbasic variable $x_j, j \in J_N$

If the optimality conditions are not holds then from a basis feasible solution and the problem in canonical form, the simplex algorithm chooses a non-basic variable that has a positive reduced cost, that is, a variable that, if increased, would increase the objective function. Then it increases the value of that variable as much as possible, without violating the non-negativity of the basic variables. That variable is made basic; (at least) one of the old basic variable becomes 0, and one becomes non-basic. The sequence of operations called a pivot goes from the canonical form with respect to the old basis to the canonical form with respect to the new basis.

Let us now to explain the idea above in more detail form. So, if on the basis feasible solution the conditions of theorem (5.11) are not valid, the simplex method replaced the basis feasible solution by new one using the following formula:

$$\bar{x} = x + \theta l_s \tag{5.12}$$

where the vector $l_s \in \mathbf{R}^n$ is called the direction of the changing of the feasible solution x, and the number $\theta > 0$ is a step along this direction l_s. The feature of the simplex method is consist in a special choice of the vector l_s. Actually that choice can be defined from the geometrical point of view on the basis feasible solution and basic idea of simplex method. The basis feasible solution is a some vertex of the polyhedral set (5.3). The iteration of the simplex method represents the movements along the edges of X or vertex to vertex movements which would increase the value of the objective function.

In analytical form, the problem of the construction of iteration reduced to the problem of the construction of elements l and θ from (5.12).

First, we start with the construction of l.

Since the simplex method is exact method, then $x, \bar{x}$ are feasible solutions. Then the direction l should be admissible. Let X - the set of feasible solutions. We will say that l-admissible direction at the point x with respect to the set X, if there is the number $\theta_0 > 0$ such that $x + \theta l \in X,\ \forall \theta \in [0, \theta_0]$.

It is easy to see that the set of admissible directions at the point x represents some cone $K_{adm}(x|X)$, i.e. $l \in K_{adm}(x|X)$, then $\theta l \in K_{adm}(x|X)$ for $\forall \theta \geq 0$. Also, it is clear that this cone is not bounded set. Since, on each iteration the direction l should belong to the cone $K_{adm}(x|X)$ and, in addition, this direction should be chosen such that on each iteration the function $c^{'}l$ achieves the maximum value on the simplex normed set:

$$N_s = \{l \in \mathbf{R}^{n-m} : \sum_{j \in J_N} |l_j| \leq 1\} \tag{5.13}$$

Choice of nonbasic to move The vector l_s has one non-zero non-basis component $l_{sj_0}, |l_{sj_0}| = 1, j_0 \in J_N$ and by this reason in (5.12) changes only one x_{j_0} of non-basis components of feasible solution x. The index $j_0 \in J_N$ is defined from the following condition:

$$|\Delta_{j_0}| = \max |\Delta_j|, J \in J_N(x) = \left\{ j \in J_N : \begin{array}{ll} \Delta_j < 0, & \text{for} \quad x_j = d_{*j}; \\ \Delta_j > 0, & \text{for} \quad x_j = d^*_j \end{array} \right\}. \tag{5.14}$$

The basis component of l_s is constructed by formula $l_B = -A_B^{-1} A_N l_N, l_N = -e_{j_0} sgn \Delta_{j_0}$. The search for j_o is called *Pricing*.

Steplength In (5.12) the step length θ along $l_s$computed above should be chosen without violating the simple constraints in order to improve the objective as much as possible

$$d_* \leq x + \theta l_s \leq d^* \tag{5.15}$$

on the vector $x(\theta) = x + \theta l_s$. Denote by θ_j the maximal step length determined by $j-$th constraint of (5.15). For each j it is possible only three cases:

1. $l_j > 0$, the component $x_j(\theta)$ increases and achieves the critical value d_j^* with $\theta = \theta_j = \frac{d_j^* - x_j}{l_{sj}}$;
2. $l_j < 0$, the function $x_j(\theta)$ decreases and achieves the critical value d_{*j} with $\theta = \theta_j = \frac{d_{*j} - x_j}{l_{sj}}$;
3. $l_j = 0$, the component $x_j(\theta)$ does not change $x_j(\theta) = x_j$, i.e. we can put $\theta_j = \infty$.

Thus we have the following formula for the step length

$$\theta_j = \begin{cases} \frac{d_j^* - x_j}{l_{sj}}, & for\ l_j > 0, \\ \frac{d_{*j} - x_j}{l_{sj}}, & for\ l_j < 0, \\ \infty, & for\ l_j = 0, j \in J_B \cup j_0. \end{cases} \tag{5.16}$$

The maximal admissible step length θ is equal to

$$\theta = \min\{\theta_{j_0}, \theta_B\} \tag{5.17}$$

$$\text{where}\ \ \theta_B = \theta_{j^*} = \min \theta_j,\ j \in J_B. \tag{5.18}$$

This is often called *the ratio test.*

If $\theta = \theta_{j_0}$ then the feasible solution $\bar{x}$ is a basis feasible solution with old basis set J_B and the conditions of (5.11) by index $j_0 \in J_N$ are holds.

If the conditions (5.11) are valid for $j \in J_N \setminus j_0$ then $\bar{x}$ is optimal feasible solution. Otherwise, we should find the new index $\bar{j} \in J_N \setminus j_0$ without changing the basis set J_B and continue the operations mentioned above.

Basis change Let $\theta = \theta_B = \theta_{j^*}$. In this case we have

$$d_{*j_0} < x_{j_0} < d_{j_0}^*$$

and

$$\bar{x}_{j_*} = x_{j^*} + \theta_{j^*} l_{sj^*} = \begin{cases} x_j + \frac{d_{j^*}^* - x_j}{l_{sj^*}} l_{sj^*}, & \text{for } l_{sj^*} > 0, \\ x_j + \frac{d_{*j^*} - x_j}{l_{sj^*}} l_{sj^*}, & \text{for } l_{sj^*} < 0, \end{cases} = \begin{cases} d_{j^*}^*, & \text{for } l_{sj^*} > 0, \\ d_{*j^*}, & \text{for } l_{sj^*} < 0. \end{cases} \tag{5.19}$$

The new basis set $\bar{J}_B = (J_B \setminus j^*) \cup j_0$ is obtained by eliminating from J_B the index j^* and adding the index j_0. Since $l_{sj^*} \neq 0$, then after replacing the columns a_{j^*} by a_{j_0} in the matrix A_B we will get the non-degenerated matrix $\bar{A}_B = A(I, \bar{J}_B)$. All components of $\bar{x}_j$ lies on the boundaries d_j^* or d_{*j} $j \in \bar{J}_N, |\bar{J}_N| = n - m$, by construction. Hence $\bar{x}$ is the basis feasible solution and the iteration of the simplex method is stopped. The basis is *updated*, and all steps are repeated.

The cost function is increased on the value $|\Delta_{j_0}|\theta > 0$ on iteration of simplex method, while $\theta > 0, |\Delta_{j_0}| > 0$, by construction, in the case of non-degeneracy of the basis feasible solution $\bar{x}$.

The operations above are called the second phase of the simplex method. And the collection of the operations to construct the initial basis feasible solution or to establish the unsolvability of the problem (5.2) are constitutes the first phase of the simplex algorithm. Let $x^* \in \mathbf{R}^n, x_j^* = d_{*j} \vee d_j^*, j \in J; d_a = (d_i, i \in I) = b - Ax^*; x_a = (x_{n+i}, i \in I)$. Consider the auxiliary problem:

$$\sum_{i=1}^{m} x_{n+i} \to \min, \tag{5.20}$$
$$A(i,J)x + x_{n+i} sgnd_i = b_i, \quad i = 1,2,...,m;$$
$$d_* \le x \le d^*, \ 0 \le x_{n+i} \le |d_i|, \quad i = 1,2,...,m,$$

which is called the problem of the first phase of the simplex method.

The variables $x_{n+i}, i = 1,2,...,m$ added to a linear program in phase 1 to aid finding a feasible solution are called an artificial variables. The problem (5.20) always has a solution since the lower bound of the feasible solution and the objective function is 0. The main purpose of the first phase of simplex method is "to destroy" artificial variables or other words to transform them to zero.

Theorem 2. *The set of feasible solution of (5.2) is not empty iff the optimal solution* $(\breve{x}, \breve{x}_a)$ *of (5.20) has the following property:*

$$\breve{x}_a = 0. \tag{5.21}$$

Reduce the problem (5.20) to canonical form and solve it, starting from the basis feasible solution $(x^*, x_a^* = d_a^*)$, where $d_a^* = (|d_i|, i = 1,2,..,m)$. The basis set consist of the artificial indices $J_B = J_a = \{n+i, i \in I\}$, and the columns of the basis matrix A_B represented by the positive or negative linear independent unit m- vectors.

After first phase of simplex method are possible the following outcome:

1. $\breve{x}_a \neq 0$;
2. $\breve{x}_a = 0,$ among the basis indices of the optimal solution there are no the artificial indices;
3. $\breve{x}_a = 0,$ among the basis indices of the optimal solution there are the artificial indices;

Analyze these possibilities: In a Case 1) the solution process for the initial problem (5.2) is stopped because its constraints are inconsistent.
In a Case 2) the vector $\breve{x}$ is a basis feasible solution of the original problem (5.2).

And we can start the second phase of simplex method with initial feasible solution $\breve{x}$.

In Case 3) we should construct the buffer problem in order to realize the main purpose of the first phase of simplex method:

$$\begin{aligned} c^{'}x \to \max, \\ A(i,J)x + x_a(I_1) = b(I_1), \quad i = I_1; \qquad A(I_2,J)x = b(I_2); \\ d_* \le x \le d^*,\ 0 \le x_{n+i} \le 0, \quad i \in I_1 \end{aligned} \tag{5.22}$$

where $I_1 = \{i \in I : n+i \in \breve{J}_B\}, I_2 = I \setminus I_1$, $\breve{J}_B$ is a basis index set after first phase. The problem (5.22) is equivalent to the (5.2). And to get the optimal solution of it used the simplex method.

Remark 3. *All non-basis components of the basis feasible solution for the problem (5.1) are equal to zero. And the basis components of the non-degenerated basis feasible solution are positive. From the theorem 2 follows the classical optimality criteria.*

Theorem 3. *For the optimality of a basis feasible point x of the problem (5.1) it is sufficient and, in the case of non-degeneracy of it, also necessary, that the all non-basis estimates were non-negative:*

$$\Delta_j \ge 0, j \in J_N. \tag{5.23}$$

Remark 4. *For the problem (5.1) after first phase of the simplex method it is not necessary to formulate the buffer problem. Before to goes to the second phase we need to change the artificial indices using special rules. (other words ,to perform the substitution of the basis).*

5.2 Adaptive method

In Phase I a starting basic feasible solution is sought to initiate Phase II or to determine that no feasible solution exists. If found, then in Phase II an optimal basic feasible solution or a class of feasible solutions is sought.

Since the simplex method among the various types of feasible solutions is used only the very specific basic feasible solution.

However, it is hard to expect that starting feasible solution given by experts or obtained from the practical experience contained the same number of the non-zero components as much the constraints in the problem. Usually, the number of these components is larger and by this reason the feasible solution is not basic. The simplex method is not used the non-basic feasible solution, other words it not take into account the knowledge of the people working on the practical problem mathematically described by (5.1). This disadvantage was already mentioned in [Danzig, Hass, Bill]

Actually the first peculiarity of the adaptive method is that instead of the basis feasible solution it is used the notion of a support feasible solution, which haven't the disadvantage of the basic feasible solution described above.

Introduce the basic notion of the adaptive method for the problem (5.2).

Definition 4. *The set of indices $J_{supp} \subset J$, $|J_{supp}| = m$ is called a support if the $m \times m$-matrix $A_{supp} = A(I, J_{supp})$ is non-singular.*

The support J_{supp} is a minimal set of indexes of J such that for any choice of the vector b and non-support components $x_N = (x_j, j \in J_N)$, $J_N = J \setminus J_{supp}$, (where the subscript N implies "unsupported"), the general constraints can be satisfied by choosing the support components $x_{supp} = (x_j, j \in J_{supp})$. Really, the general constraints $Ax = b$ in component form can be expressed as

$$A_{supp}x_{supp} + A(I, J_N)x_N = b. \tag{5.24}$$

Therefore, to satisfy the equality it is sufficient to set

$$x_{supp} = A_{supp}^{-1}(b - A_N x_N), A_N = A(I, J_N) \tag{5.25}$$

Also, it easy to see that if we put $b = 0$ and $x_N = 0$ then (5.24) holds iff $x_{supp} = 0$.

It is clear that inverse matrix with regard to the support plays a special role.

In the iteration method the support will be changed together with feasible points. Therefore the main object of transformation is a pair comprising a feasible point and a support.

Definition 5. *The pair $\{x, J_{supp}\}$ constitutes by an any feasible point and an any support will be called a support feasible (SF) point. The SF-point will be called non-degenerate if*

$$d_{*j} \leq x_j \leq d_j^*,\ j \in J_{supp}.$$

The support feasible solution is not denied the basis. In some sense it can be considered as a generalization of the basis in simplex method. The generalization of basis feasible solution is carry out in a such a way that algorithm constructed on a SF-points has the advantages of simplex method (simplicity, finiteness, etc.)

Remark 5. • *Noting the classic simplex-method for LP optimization, it can be shown that the introduced matrix A_{supp} is composed, in fact, by the vectors of the general constraints of (5.2) which forms the basis for the corresponding basis feasible plan of the form $x = \{x_B, x_N\}$ where $x_N = (x_j = 0,\ j \in J_N)$ and $x_{supp} = A_{supp}^{-1} b$.*

- *The proposed method has been extended for the interval problem of LP with two-sided constraints*

$$c^T x \to \max,$$
$$b_* \le Ax \le b^*,$$
$$d_* \le x \le d^*.$$

- *In general case (see, for example, the original paper [2]) the support notion was introduced as follows: Let $I_{supp} \subset I$ and $J_{supp} \subset J$ are nonempty indexes sets such as $|J_{supp}| = |I_{supp}|$. The couple $\{I_{supp}, J_{supp}\}$ is called a support if the $|I_{supp}| \times |J_{supp}|$-submatrix $A_{supp} = A(I_{supp}, J_{supp})$ is non-singular. Represent now the matrix A in block form as*

$$A = \begin{pmatrix} A_{supp,supp} & A_{supp,N} \\ A_{N,supp} & A_{N,N} \end{pmatrix} \tag{5.26}$$

where $A_{supp,supp} = A(I_{supp}, J_{supp})$, $A_{supp,N} = A(I_{supp}, J_N)$, $A_{N,supp} = A(I_N, J_{supp})$ and $A_{N,N} = A(I_N, J_N)$, $I_N = I \setminus I_{supp}$. Then the corresponding relations of (5.24)-(5.25) are rewritten as follows

$$A_{supp,supp} x_{supp} + A_{supp,N} x_N = b_{supp},\ b_{supp} = (b_i, \in I_{supp}) \tag{5.27}$$
$$A_{N,supp} x_{supp} + A_{N,N} x_N = b_N,\ b_N = (b_i,\ i \in I_N)$$

and

$$x_{supp} = A_{supp,supp}^{-1}(b - A_{supp,N} x_N) \tag{5.28}$$

respectively.

The support feasible solution $x = \{x_{supp}, x_N\}$,in contrast to the basis feasible solution $x = \{x_B, x_N\}$, has not necessary zero(trivial) non-support components $x_j, j \in J_N$.

Example Consider the notion of support on more general problem, then (5.2):

$$\begin{aligned} F = 2x_1 + x_2 &\to \max \\ 2 \le x_1 + 2x_2 &\le 6 \\ -2 \le -x_1 + x_2 &\le 1 \\ 1 \le x_1 \le 3, \qquad 0 \le x_2 &\le 2 \end{aligned} \tag{5.29}$$

In accordance to out notation we will have:

$$\begin{aligned} c^T x &\to \max, \\ b_* \le Ax &\le b^*, \\ d_* \le x &\le d^*. \end{aligned}$$

where

$$A = \begin{pmatrix} 1 & 2 \\ -1 & 1 \end{pmatrix} = A(I, J)$$

$$\begin{aligned} I = \{1,2\}, \quad J = \{1,2\}, \\ x = (x_1, x_2) = x(J) \\ b_* = (2; -2) = b_*(I) \quad b^* = (6; 1) = b^*(I) \\ d^* = (3; 2) = d^*(J), \quad d_* = (1; 0) = d_*(J) \\ c = (2; 1) = c(J). \end{aligned}$$

In accordance with remark 5 the definition 4 can be extended for the problem (5.29) as following:

1. $\left(I^1_{supp} = \varnothing\ ,\ J^1_{supp} = \varnothing\right)$ *since* $\det A^1_{supp}(I^1_{supp}, J^1_{supp}) \neq 0$
2. $\left(I^2_{supp} = \{1\},\ \ J^2_{supp} = \{1\}\right)$ *since* $\det A^2_{supp}(I^2_{supp}, J^2_{supp}) = \det(1) \neq 0$
3. $\left(I^3_{supp} = \{1\},\ \ J^3_{supp} = \{2\}\right)$ *since* $\det A^3_{supp}(I^3_{supp}, J^3_{supp}) = \det(2) \neq 0$
4. $\left(I^4_{supp} = \{2\},\ \ J^4_{supp} = \{1\}\right)$ *since* $\det A^4_{supp}(I^4_{supp}, J^4_{supp}) = \det(-1) \neq 0$
5. $\left(I^5_{supp} = \{2\},\ \ J^5_{supp} = \{2\}\right)$ *since* $\det A^5_{supp}(I^5_{supp}, J^5_{supp}) = \det(1) \neq 0$
6. $\left(I^6_{supp} = \{1,2\},\ \ J^6_{supp} = \{1,2\}\right)$ *since* $\det A^6_{supp}(I^6_{supp}, J^6_{supp}) = \det \begin{pmatrix} 1 & 2 \\ -1 & 1 \end{pmatrix} \neq 0$

Thus we have 6 possible supports. The support number 1 is called empty support.

Thus, the support J_{supp} is a minimal set of indices of J such that for any choice of the vector b and components $x_N = (x_j, j \in J_N)$, $J_N = J \setminus J_{supp}$, the general constraints can be satisfied by choosing the components $x_{supp} = (x_j, j \in J_{supp})$.

This fact can be effectively used for the search of support sets for practical implementation since usually experts are able to indicate the most critical equalities that can be obligatory to satisfy.

Next in order to formulate the optimality criteria consider the formula of the objective value increment.

5.2.1 Objective value increment formula

Assume that the initial SF-point is known. Examine the behavior of the objective function when the feasible point is changed. Besides the SF-point $\{x, J_{supp}\}$ we consider an arbitrary n-vector $\bar{x}$ satisfying the constraints $A\bar{x} = b$. Such vectors are called a pseudo-feasible points. We set $\Delta x = \bar{x} - x$ and calculate the increment of the objective value

$$\Delta F(x) = F(\bar{x}) - F(x) = c^T\bar{x} - c^T x = c^T \Delta x \tag{5.30}$$

It is obvious that

$$A\Delta x = A(\bar{x} - x) = A\bar{x} - Ax = b - b = 0 \tag{5.31}$$

or in component form

$$A_{supp}\Delta x_{supp} + A_N \Delta x_N = 0 \tag{5.32}$$

and

$$\Delta x_{supp} = -A_{supp}^{-1} A_N \Delta x_N. \tag{5.33}$$

Thus for any $\Delta x_N = (\Delta x_j,\ j \in J_N)$ and Δx_{supp} defined by (5.33) we obtained the vector $\Delta x = (\Delta x_{supp}, \Delta x_N)$ such that $\bar{x} = x + \Delta x$ satisfies the general constraints.

Substituting the vector Δx into (5.30) we get

$$\Delta F(x) = c_{supp}^T \Delta x_{supp} + c_N^T \Delta x_N = -(c_{supp}^T A_{supp}^{-1} A_N - c^T)\Delta x_N. \tag{5.34}$$

The vector

$$\Delta^T = (\Delta_j, j \in J) = c^T_{supp} A^{-1}_{supp} A - c^T \tag{5.35}$$

will be called a support gradient. It is evident its support components are equal to zero:

$$\Delta^T_{supp} = c^T_{supp} A^{-1}_{supp} A_{supp} - c^T_{supp} = c^T_{supp} - c^T_{supp} = 0.$$

To calculate the non-support components of the support gradient it is convenient to use the vector of multipliers or potential vector:

$$u^T = c^T_{supp} A^{-1}_{supp}. \tag{5.36}$$

Then we have

$$\Delta^T_N = u^T A_N - c^T_N \tag{5.37}$$

or

$$\Delta_j = u^T a_j - c_j,\ j \in J_N \tag{5.38}$$

where $a_j = A(I, j)$ is the j-th column of the matrix A. From (5.34)-(5.35) we get the follows

$$\Delta F(x) = -\Delta^T_N \Delta x_N = -\sum_{j \in J_N} \Delta_j \Delta x_j. \tag{5.39}$$

The physical sense of the support gradient Δ can be explained using (5.39). Let

$$\Delta x_N = (0, ..., 0, \Delta x_k, 0, ..., 0), \tag{5.40}$$

where $k \in J_N$ is a certain index. The component Δx_{supp} are found from (5.33):

$$\Delta x_{supp} = -A^{-1}_{supp} A_N \Delta x_N = -A^{-1}_{supp} a_k \Delta x_k. \tag{5.41}$$

According to (5.39)

$$\Delta F(x) = -\Delta_k \Delta x_k, \quad k \in J_N. \tag{5.42}$$

Remark 6. *From mathematical analysis it is known that if for the function $f(x), x \in \boldsymbol{R}^n$ has been proved the following formula:*

$$\Delta f(x) = f(x + \Delta x) - f(x) = a^{'} \Delta x + o(\|\Delta x\|) \tag{5.43}$$

then $a = \frac{\partial f(x)}{\partial x} = gradf(x)$ is a gradient of the function $f(x), x \in \boldsymbol{R}^n$ at the point x. In mechanics the vector a expresses the rate of change of the function $f(x), x \in \boldsymbol{R}^n$ at the point x.

The formula (5.42) has been obtained for the (5.40), i.e. for the ($\Delta x_{j_k} \neq 0, j_k \in J_N; \Delta x_j = 0, j \in J_N, j \neq j_k$).

Taking into account the remark above we can concludes that Δ_k is the rate of change of the objective function taken with opposite sign when the k-th non-support component of the feasible point x is increased and all the non-support components (besides the k-th) are fixed. At the same time the support components are changed in a such way to satisfy the support general constraints.

Also it should be noted that k-th components of the "classical" gradient of the objective function is equal to $\frac{\partial F}{\partial x_k} = c_k$. If so in our case the "reduced" gradient of the objective function by x is equal $\Delta_k = c_k - A^{'}u$, calculated under conditions of fulfilment of the general constraints. And these constraints are satisfied with help of the support.

Remark 7. *Another words we can get Δ_k from c_k by correction of $A^{'}u$, which depends on the matrix of general constraints. And correcting multiplier(potential) u constructed with help of support.*

Therefore Δ_k can be called a support gradient.

5.2.2 The optimality criteria

Now let x be a feasible solution. And answer on the following question: is it optimal point? To answer on this question will use the support J_{supp} and calculate the support gradient (5.38) at the SF-point $\{x, J_{supp}\}$.

Theorem 4. *For the optimality of a feasible point x it is sufficient and, in the case of non-degeneracy of SF-point $\{x, J_{supp}\}$, also necessary, that the following conditions:*

$$\begin{cases} \Delta_j \geq 0 \ \ \textit{for } x_j = d_{*j}, \\ \Delta_j \leq 0 \ \ \textit{for } x_j = d_j^*, \\ \Delta_j = 0 \ \ \textit{for } \ d_{*j} \leq x_j \leq d_j^*, \quad j \in J_N \end{cases} \tag{5.44}$$

holds.

The proof of Theorem4 can be obtained using (5.39).

Definition 6. *The pair $\{x^o, J^o_{supp}\}$ satisfying the relations (5.44) will be called an optimal SF-point.*

Remark 8. *Optimality criteria for supporting feasible solution (5.44) is differ then optimality criteria for basis feasible solution (5.11), while in the latter there is no equality condition. In Simplex method all non-basic variables $x_j, j \in J_N$ takes the limits value.*

Another significant features of the adaptive method is that on each iteration it can estimate the deviation of the support feasible solution from optimal one by the value of the objective function, while simplex method have not such possibility.

5.2.3 The suboptimality criteria

In applied problems it is very often sufficient to have a suboptimal feasible points. For this reason the problem of identification of ϵ-optimal feasible points appears. In the set of the pseudo-feasible points (i.e. the vectors of $\bar{x}$ satisfying the general constraints $A\bar{x} = b$) we consider the subset consisting of the vectors $\bar{x}$, non-support components of which satisfy the simple constraints

$$d_{*j} \le \bar{x}_j = x_j + \Delta x_j \le d_j^*,\ j \in J_N. \tag{5.45}$$

The maximal value of (5.39) under constraints (5.45) is equal

$$\max \Delta F(x) = \max_{\substack{d_{*j} - x_j \le \Delta x_j \le d_j^* - x_j, \\ j \in J_N}} (-\sum_{j \in J_N} \Delta_j \Delta x_j) = \tag{5.46}$$
$$= \sum_{j \in J_N} \left(\max_{d_{*j} - x_j \le \Delta x_j \le d_j^* - x_j} (-\Delta_j \Delta x_j) \right) =$$
$$\sum_{\substack{\Delta_j > 0, \\ j \in J_N}} \Delta_j (x_j - d_{*j}) + \sum_{\substack{\Delta_j < 0, \\ j \in J_N}} \Delta_j (x_j - d_j^*)$$

The value

$$\beta = \beta(x, J_{supp}) = \sum_{\substack{\Delta_j > 0, \\ j \in J_N}} \Delta_j (x_j - d_{*j}) + \sum_{\substack{\Delta_j < 0, \\ j \in J_N}} \Delta_j (x_j - d_j^*) \tag{5.47}$$

will be called a suboptimality estimate of the SF-point $\{x, J_{supp}\}$. The value β is a suboptimality estimates of feasible point x since this estimate is calculated in presence of particular simple constraints (5.45) on non-supporting components only (in absence of the general and simple supporting constraints on the feasible plans). It is evidently that in this case the exact estimate may be less only

$$F(x^o) - F(x) \leq \beta(x, J_{supp}) \tag{5.48}$$

The suboptimality estimates is finite if:

- $d_{*j} > -\infty, \quad$ for $\Delta_j > 0$;
- $d^*_j < +\infty, \quad$ for $\Delta_j < 0$.

We will consider only the problem when $\beta(x, J_{supp}) < \infty$.

The following theorem can be treated as a ground for the number $\beta(x, J_{supp})$.

Theorem 5. *Let us given $\epsilon \geq 0$. For a feasible point x to be ϵ-optimal it is sufficient that there exists a support J_{supp} such that*

$$\beta(x, J_{supp}) \leq \epsilon.$$

The proof follows from (5.48).

5.2.4 The dual problem. The elements of the dual theory.

It is well known that duality theory presents a powerful method for investigation of extremal problems. And using some dual elements in the developed numerical methods images, in fact, their efficiency. In this section we give a short overview of duality results applied to the considered problems.

Let $\{x^o, J^o_{supp}\}$ be an SF-point satisfying the optimality criteria (5.44)(such points can be constructed by the algorithm described below). Let u^o be a vector of multipliers (potential vector) $u^{oT} = c^T_{supp}A^{-1}_{supp}$ corresponding to the support J^o_{supp}. Next using (5.38),(5.44) we get

$$\begin{aligned} a_j^T u^o - c_j \geq 0 \quad & \text{for } x_j = d_{*j}, \\ a_j^T u^o - c_j \leq 0 \quad & \text{for } x_j = d^*_j, \\ a_j^T u^o - c_j = 0 \quad & \text{for } d_{*j} < x_j < d^*_j,\ j \in J_N \end{aligned} \tag{5.49}$$

Since $\Delta_j = 0, j \in J^o_{supp}$ we have

$$a_j^T u^o - c_j = 0, \quad j \in J^o_{supp} \tag{5.50}$$

Next, introduce the vector

$$\delta^o = \delta^o(J) = A^T u^o - c. \tag{5.51}$$

¿From the (5.50) we have the following

$$\delta^o_{supp} = 0, \quad \delta^{oT}_N = c^T_{supp} A^{-1}_{supp} A_N - c^T_N \tag{5.52}$$

Also introduce another vectors v^o, w^o with

$$\begin{aligned} v^o_j &= \delta^o_j, \ w^o_j = 0 \ \text{ for } \delta^o_j \geq 0; \\ v^o_j &= 0, \ w^o_j = -\delta^o_j \ \text{ for } \delta^o_j < 0, \ j \in J. \end{aligned} \tag{5.53}$$

According to (5.52)-(5.53) the collection $\lambda^o = (y = u^o, v = v^o, w = w^o)$ satisfies the following relations

$$A^T y - v + w = c, \ v \geq 0, \ w \geq 0. \tag{5.54}$$

Using the stated above equalities we calculate for the collection $\lambda^o = (u^o, v^o, w^o)$ the value (the value will be denoted as $\Phi(\lambda^o)$) of the following expression

$$\begin{aligned} \Phi(\lambda^o) &= b^T u^o - d_*^T v^o + d^{*T} w^o = \\ &= c^T_{supp} A^{-1}_{supp} b - \sum_{\delta^o_j \geq 0, j \in J_N} d_{*j} \delta^o_j - \sum_{\delta^o_j < 0, j \in J_N} d^*_j \delta^o_j = \\ &= c^T_{supp} A^{-1}_{supp} b - \sum_{j \in J_N} x^o_j \delta^o_j = c^T_{supp} A^{-1}_{supp} b - (c^T_{supp} A^{-1}_{supp} A_N - c^T_N) x^o_N = \\ &= c^T_{supp}(A^{-1}_{supp} b - A^{-1}_{supp} A_N x^o_N) + c^T_N x^o_N = c^T_{supp} x^o_{supp} + c^T_N x^o_N = c^T x^o = F(x^o). \end{aligned} \tag{5.55}$$

Thus

$$\Phi(\lambda^o) = F(x^o). \tag{5.56}$$

Let now $\lambda = (y, v, w)$ be an arbitrary collection of the vectors satisfying the conditions (5.54), and x be an arbitrary feasible point for the problem (5.2). And calculate the value of (5.55) for λ:

$$\begin{aligned} \Phi(\lambda) = b^T y - d_*^T v + d^{*T} w &\geq x^T A^T y - x^T v + x^T w = \\ &= x^T (A^T y - v + w) = c^T x = F(x). \end{aligned}$$

Hence

$$\Phi(\lambda) \geq F(x) \tag{5.57}$$

for any feasible point x and, in particular, for the optimal point x^o:

$$\Phi(\lambda) \geq F(x^o) \tag{5.58}$$

Taking into account (5.56)-(5.54) and the evident inequality $F(x^o) \geq F(x)$ we have

$$\Phi(\lambda) \geq \Phi(\lambda^o) = F(x^o) \geq F(x). \tag{5.59}$$

This means that $\lambda^o = (y^o, v^o, w^o)$ is an optimal solution of the following optimization problem:

$$\begin{aligned} \Phi(\lambda) = b^T y - d_*^T v + d^{*T} w \to \min, \\ A^T y - v + w = c,\ v \geq 0,\ w \geq 0. \end{aligned} \tag{5.60}$$

The formulated problem (5.60) presents an linear programming (LP) problem. Note that the both problem (5.2) and (5.60) problems are formed by the same parameters $\{c, A, b, d_*, d^*\}$.

The problem (5.2) will be called primal linear programming problem, and (5.60) will be called the corresponding dual linear programming problem.

Definition 7. *A collection $\lambda = (y, v, w)$ satisfying to all constraints of the dual problem (5.60) is called a dual feasible point. The solution $\lambda^o = (y^o, v^o, w^o)$ of the problem (5.60) is an optimal if $\Phi(\lambda) \geq \Phi(\lambda^o)$ for all dual feasible points λ.*

Theorem 6. *The problem (5.2) has a solution x^o if and only if the dual problem (5.60) has a solution λ^o such that*

$$\Phi(\lambda^o) = F(x^o),$$

i.e. the values of the objective functions on the optimal solutions of the dual and the primal problems are equal.

Note that in contrast to primal LP problem, the feasible vectors for dual LP problem can be easy constructed. Indeed, let y be an arbitrary m- vector. Calculate the vector $\delta = A^T y - c$, which will be called a co-point vector. Using the co-point vector δ we construct the auxiliary vectors $v_{(coord)}, w_{(coord)}$ as the following :

$$\begin{aligned} v_{(coord)j} = \delta_j,\ w_{(coord)j} = 0 \ \text{ for } \delta_j \geq 0; \\ v_{(coord)j} = 0,\ w_{(coord)j} = -\delta_j \ \text{ for } \delta_j < 0,\ j \in J. \end{aligned} \tag{5.61}$$

It easy to see that the collection $\lambda_{(coord)} = (y, v_{(coord)}, w_{(coord)})$ defined by (5.61) is a dual feasible point. Sometimes the condition (5.61) is called as a coordinated condition, and the corresponding collection $\lambda_{(coord)}$ given by (5.61) is called then as coordinated dual feasible point.

Remark 9. *Also the following corollary from Theorem 6 plays an important role in LP: Let x^o is optimal solution of the primal problem, and $\lambda^o = (y^o, v^o, w^o)$ is an dual optimal solution, and the vector $\delta^o = c - Ay^o$ is constructed, then the following relations are true:*

$$\begin{aligned} x_{*j} &> d_{*j} \quad \text{for } v_j^o = 0, \\ x_j^o &< d_j^* \quad \text{for } w_j^o = 0, \end{aligned} \tag{5.62}$$

and vice versa

$$\begin{aligned} v_j^o &> 0 \quad \text{then} \quad x_j^o = d_{*j} \\ w_j^o &> 0 \quad \text{then} \quad x_j^o = d_j^*, \quad j \in J. \end{aligned} \tag{5.63}$$

The relations above are equivalent to the equalities:

$$\begin{aligned} (x_j^o - d_{*j})v_j^o &= 0, \\ (d_j^* - x_j^o)w_j^o &= 0, \quad j \in J \end{aligned} \tag{5.64}$$

which are called a complementarity conditions.

The introduced dual feasible points possesses the following important extremal property.

Lemma 1. *Let $\lambda_{(coord)} = (y, v_{(coord)}, w_{(coord)})$ be a coordinated dual feasible point and $\bar{\lambda} = (y, \bar{v}, \bar{w})$ be arbitrary (uncoordinated) feasible point. Then $\Phi(\lambda) \le \Phi(\bar{\lambda})$.*

In other words, the lemma says that the optimal solution of dual optimization problem could be found in the set of the dual coordinated feasible points, in fact.

In the set of coordinated dual feasible points define the so-called accompanying dual feasible points that play an important role in the developed next decomposition of suboptimality estimate and iteration procedure, in general.

For an arbitrary support J_{supp} construct the vector of multipliers $u^T = c_{supp}^T A_{supp}^{-1}$ and the vector $\delta = A^T u - c$.

The coordinated dual point will be called accompanying dual feasible point if $y = u$: $\lambda_{(acc)} = (u, v_{(coord)}, w_{(coord)})$.

Thus, the co-point corresponding to the accompanying dual feasible point coincides with the support gradient, i.e. $\delta = \Delta$. Sometimes the vector u is called as a vector of Lagrange multipliers accompanying the support J_{supp}, and the vector δ is then called as co-point vector accompanying the support J_{supp}.

Let J_{supp} be a support and Δ be the correspondent support gradient. The vector $z = z(J) = (z_{supp}, z_N)$ with

$$\begin{aligned} z_j &= d_{*j} \ \text{ if } \Delta_j > 0, \\ z_j &= d_j^* \ \text{ if } \Delta_j < 0, \\ z_j &= d_{*j} \ \text{ or } d_j^* \ \text{ if } \ \Delta_j = 0,\ j \in J_N; \\ z_{supp} &= A_{supp}^{-1}(b - A_N z_N) \end{aligned} \tag{5.65}$$

will be called an accompanying pseudo-feasible point.

Note, if z is an accompanying pseudo-feasible point then $Az = b$ since multiplying the last equality in (5.65) by A_{supp} we have $A_{supp}z_{supp} = b - A_N z_N$ and hence $A_{supp}z_{supp} + A_N z_N = Az = b$.

Setting $z_j = d_{*j}$ for $\delta_j = 0$ we believe that δ_j is infinitesimal positive number, i.e. $\delta_j = +0$; and in the case $z_j = d_j^*$ for $\delta_j = 0$ we believe that δ_j is infinitesimal negative. i.e. $\delta_j = -0$. Also, denote the index sets by

$$\begin{aligned} J_N^+ &= \{j \in J_N : z_j = d_{*j}\} \\ J_N^- &= \{j \in J_N : z_j = d_j^*\}. \end{aligned} \tag{5.66}$$

Note that the constructed accompanying pseudo-feasible point $z = z(J) = (z_{supp}, z_N)$ is an optimal solution of the following reduced problem:

$$\begin{gathered} c^T x \to \max \\ Ax = b, \quad d_{*N} \le x_N \le d_N^*. \end{gathered} \tag{5.67}$$

This can be easy stated since the conditions (5.65) coincides with the optimality conditions (5.44) re-written for the given optimization problem.

In addition, if the following inequalities

$$d_{*supp} \le x_{supp} \le d_{supp}^*, \tag{5.68}$$

hold, then the constructed vector z is a feasible point of the primal problem (5.2), and, hence, this vector is an optimal solution for the problem (5.2).

From (5.47) follows that the vector $\Delta x_N = z_N - x_N$ maximizes the increment $\Delta F(x)$ of the objective value and the suboptimality estimate in this case is given as

$$\beta(x, J_{supp}) = \Delta_N^T(x_N - z_N). \tag{5.69}$$

Thus, we show that for any support J_{supp} the following equality is fulfilled

$$c^T z = \Phi(\lambda_{(acc)}), \tag{5.70}$$

where $\lambda_{(acc)}$ is the dual feasible point accompanying the support J_{supp} and z is an accompanying pseudo-feasible point.

5.2.5 Decomposition of the suboptimality estimate. Degrees of non-optimality of the feasible point and support.

Let $\{x, J_{supp}\}$ be an SF-point, and $\beta(x, J_{supp})$ be its suboptimality estimate calculated by (5.47), and $\lambda_{(acc)} = (y, v, w)$ be an accompanying dual feasible point. Then the following decomposition of estimate of $\{x, J_{supp}\}$ is valid

$$\begin{aligned}\beta(x, J_{supp}) = \Delta_N^T(x_N - z_N) = \delta^T(x - z) = (u^T A - c^T)(x - z) = c^T z - c^T x = \\ = \Phi(\lambda_{(acc)}) - F(x) = \Phi(\lambda) - \Phi(\lambda^o) + F(x^o) - F(x) = \\ = \beta(J_{supp}) + \beta(x),\end{aligned} \tag{5.71}$$

Thus

$$\beta(x, J_{supp}) = \beta(x) + \beta(J_{supp}) \tag{5.72}$$

where $\beta(x) = F(x^o) - F(x)$ means the deviation of the current objective value $F(x)$ from the optimal ones, and $\beta(J_{supp}) = \Phi(\lambda) - \Phi(\lambda^o)$ denotes the deviation of the dual objective value function $\Phi(\lambda)$ from the optimal ones. The value $\beta(x)$ is called the degree of non-optimality of the current feasible point x. It is clear that for the optimal feasible point x^o the measure of non-optimality degree $\beta(x^o) = 0$. Analogously the $\beta(J_{supp})$ is the non-optimality degree of the current support J_{supp} . From the (5.72) follows that $\beta(x, J_{supp})$ estimates the suboptimality of the feasible point x only, if the measure of non-optimality degree of support is equal zero:

$$\beta(J_{supp}) = 0. \tag{5.73}$$

The support J_{supp} will be called optimal, if the condition (5.73) above holds.

Theorem 7. *If $x = x^\epsilon$ be an ϵ-feasible point, then there is the support J_{supp} such that the following inequality*

$$\beta(x, J_{supp}) \le \epsilon$$

holds.

5.3 Iteration of the algorithm

We assume that an initial SF-point $\{x, J_{supp}\}$ and the accuracy $\epsilon \geq 0$ are given. If the suboptimality estimate is not satisfactory $\beta(x, J_{supp}) > \epsilon$ then we should change the current SF-point to a new one $\{x, J_{supp}\} \rightarrow \{\bar{x}, \bar{J}_{supp}\}$ due to the developed iterations algorithm.

The proposed iteration method is based on the principle of decreasing of the suboptimality estimate. In other words the transformation $\{x, J_{supp}\} \rightarrow \{\bar{x}, \bar{J}_{supp}\}$ is carried out in a such way that

$$\beta(\bar{x}, \bar{J}_{supp}) < \beta(x, J_{supp}).$$

The iteration $\{x, J_{supp}\} \rightarrow \{\bar{x}, \bar{J}_{supp}\}$ is realized by two procedures:

1. transformation of the feasible point $x \rightarrow \bar{x}$, which decreases the non-optimality degree of the feasible point : $\beta(\bar{x}) \leq \beta(x)$;
2. transformation of the support $J_{supp} \rightarrow \bar{J}_{supp}$, which decreases the degree of non-optimality of the support: $\beta(\bar{J}_{supp}) \leq \beta(J_{supp})$.

5.3.1 Procedure of the changing the feasible point $x \rightarrow \bar{x}$

Let $\{x, J_{supp}\}$ be an SF-point. Calculate the support gradient $\Delta = A^T u - c$ where $u = A_{supp}^{-1^T} c_{supp}$, the non-suppport components of accompanying pseudo-feasible point z_N as

$$z_j = \begin{cases} d_{*j}, \ \text{for} j \in J_N^+ \\ d_j^*, \ \text{for} j \in J_N^- \end{cases} \tag{5.74}$$

and the suboptimality estimate

$$\beta(x, J_{supp}) = \Delta_N^T (x_N - z_N). \tag{5.75}$$

If $\beta(x, J_{supp}) \leq \epsilon$, then x is an ϵ-optimal feasible point and the solution procedure is stopped.

Let $\beta(x, J_{supp}) > \epsilon$. Then we construct

$$z_{supp} = A_{supp}^{-1}(b - A_N x_N)$$

If $d_{*j} \leq z_{supp} \leq d_j^*$ then the vector z is an optimal feasible point, and in this case the solution procedure is also stopped,

otherwise we are needed to transform the current feasible point x. For this purpose it is necessary to find an admissible direction in the space of considered variables along which we will construct the new feasible point $\bar{x}$. According to this principle the new feasible point $\bar{x}$ is constructed from the old one x as

$$\bar{x} = x + \theta l, \tag{5.76}$$

where the vector l is called the direction of the changing of the feasible point x, and the number $\theta \geq 0$ is a step along this direction l.

Thus the problem of the construction of iteration reduced to the problem of the construction of elements l and θ from (5.76).

First, we start with the construction of l.

Let X - the set of feasible points. We will say that l-admissible direction at the point x with respect to the set X, if there is the number $\theta_0 > 0$ such that $x + \theta l \in X,\ \forall \theta \in [0, \theta_0]$.

It is easy to see that the set of admissible directions at the point x represents some cone $K_{adm}(x|X)$, i.e. $l \in K_{adm}(x|X)$, then $\theta l \in K_{adm}(x|X)$ for $\forall \theta \geq 0$. Also, it is clear that this cone is not bounded set. Since, on each iteration the direction l should belong to the cone $K_{adm}(x|X)$ and, in addition, this direction should be chosen such that on each iteration the measure of non-optimality is decreased, i.e. the direction l should be chosen in such way that the following inequality holds

$$\beta(\bar{x}) - \beta(x) = c^T x - c^T \bar{x} = -c^T(x + \theta l) + c^T x = -\theta c^T l \leq 0 \text{ for } \ \theta > 0.$$

Hence we have

$$c^T l \geq 0. \tag{5.77}$$

The direction $l \in \mathbb{R}^n$ satisfying (5.77) will be called an increasing direction of the objective function. The set of such directions denote as $K_{incr}(x) = \{l \in \mathbb{R}^n : c^T l \geq 0\}$. Thus, the iteration procedure uses the increasing direction l as admissible directions. Such direction will be called the proper direction. The set of the proper directions denote by $K_{pr}(x|X)$. Obviously, $K_{pr}(x|X) = K_{adm}(x|X) \cap K_{incr}(x|X)$. The set $K_{pr}(x|X)$ usually contains a lot of different directions with the same length. The questions is which element we should choose for the construction of our iteration (5.76)? Since $c^T l = \frac{\partial(c^T x)}{\partial l}$ is the rate of the changing of the objective function along the direction l, then it is reasonable to take for the "fastest" decreasing of the measure of nonoptimality the direction l^o such that

$$c^T l^o = \max_{l \in K_{pr}(x|X)} c^T l \tag{5.78}$$

Clear, that in the case $x \neq x^o$ the problem (5.78) has no solution while the objective function is not bounded. In fact, if we suppose that l^o is the solution, then $c^T l^o > 0$ and

θl^o also the solution for $\forall\theta > 0$ with $\theta c^T l^o \to +\infty$, where $\theta \to +\infty$. To avoid this difficulties impose the additional constraints on the proper directions, the so-called normed condition: $l \in N$ where N is suitable determinated compact set from $\mathbb{R}^n$. Then the needed direction l is defined by the following formula:

$$c^T l^o = \max_{l \in K_{pr}(x|X) \cap N} c^T l. \tag{5.79}$$

The set N will be called a normed set.

Since the problem (5.79) is auxiliary problem, then the normed set N should be quite simple. The most simple set is the unit ball of $N = \{l \in \mathbb{R}^n : \|l\| \leq 1\}$ where $\|l\|$ is a norm of the vector l. There are different kind of norms. The most popular among them are : 1)$\|l\| = \sqrt{l^T l}$; 2) $\|l\| = \max\limits_{j \in J} |l_j|$; 3) $\|l\| = \sum\limits_{j \in J} |l_j|$. Then the corresponding set N are : $N_1 = \{l : l^T l \leq 1\}$; $N_2 = \{l : \max\limits_{j \in J} |l_j| \leq 1\}$; $N_3 = \{l : \sum\limits_{j \in J} |l_j| \leq 1\}$. It is clear that for each norm we will have the own solution l^o. Moreover, for any proper direction $l \in K_{pr}$ we can select the norm in a such way that l be the solution of the problem (5.79) with that norm. The problem on this stage is the following: How to choose the normed set for the problem (5.2)? It is known that the simplex-method based on the set N_3. The disadvantage of all proposed normed sets consists in that they are not connected to the initial problem (5.2) at all. But obviously the choice of the normed set strongly influences on the form of l^o and, hence, on the progress of the solution of (5.2). We can find an "optimal" normed set N^o if this choice will be directly connected with the problem (5.2). Indeed, let

$$N^o = \{l \in \mathbb{R}^n : x + l \in X\}. \tag{5.80}$$

Then the problem (5.79) can be rewritten as

$$c^T l \to \max, \quad A(x + l) = b, \quad d_* \leq x + l \leq d^*. \tag{5.81}$$

Let l^o is a solution of (5.81). Then $\bar{x} = x + l^o = x^o$, i.e. the transformation the point x with step $\theta = 1$ along the direction l^o lead us exactly to the optimal point and the initial problem is solved in one step of iteration. The disadvantage of the normed set (5.80) consists in that the problem of construction of the direction l^o is equivalent to the complexity of the initial problem (5.2), while after substitution $\bar{x} = x + l$ we have

$$c^T \bar{x} - c^T x \to \max_{\bar{x}}, \quad A\bar{x} = b, \quad d_* \leq \bar{x} \leq d^*.$$

In order to avoid this obstacle we propose to reduce the problem (5.81) by relaxing some constraints of that problem. Namely, we remove the following constraints: 1) the non-supporting general constraint $A(I, J_N)(x + l) = b_N$; 2) the supporting simple constraint

$d_{*supp} \leq x_{supp} + l_{supp} \leq d^*_{supp}$. Then we obtain instead of the set N^o in (5.80) the new normed set of the form

$$N_a = \{l \in \mathbb{R}^n : A(I, J_{supp})(x + l) = b_{supp},\ d_{*N} \leq x_N + l_N \leq d^*_N, \} \tag{5.82}$$

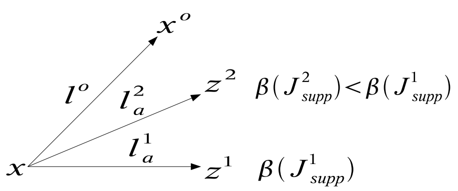
Then the problem (5.79) has the following form:

$$c^T \bar{x} - c^T x \to \max_{\bar{x}}, \quad A(I, J_{supp})\bar{x} = b_{supp}, \quad d_{*N} \leq \bar{x}_N \leq d^*_N. \tag{5.83}$$

The problem above we already met when the suboptimality estimate was derived (see subsection 5.2.3-5.2.4). The solution of this problem is the pseudo feasible point z accompanying the support J_{supp}. The "optimal" normed set N_a we get from the N^o by reducing some constraints. The set N_a can be called as suboptimal normed set.

The direction l_a obtained from the (5.83)

$$l_a = z - x$$



(5.84)

will be called an adaptive direction .

The vector z depends on the support J_{supp} and if the support J_{supp} will be better, then the corresponding vector z will be close to the optimal vector x^o. Analogously, the direction l_a will be close to the optimal direction l^o, when the support J_{supp} is close to the optimal support.

Next along the obtained direction (5.84) we find the new feasible point $\bar{x}$ among the vectors described by

$$x(\theta) = x + \theta l_a, \qquad \text{where } \theta \geq 0 \text{ is a step length along } l_a.$$

Actually the principle of calculation of the step θ is general for all exact methods: the searching procedure of the θ along the proper direction leads until the the point

$$x(\theta) = x + \theta l_a$$

belongs to the set of admissible points X.

The specific features of the proposed iteration procedure is the approach to construct the direction l_a. According to (5.84) the vector l_a is directed from x to the accompanying feasible point z, and satisfies optimality condition (5.44) with respect to the non-supporting components, i.e. it is a solution of (5.69)

Also it has the property $\beta(z, J_{supp}) = 0$ and the components z_N coinciding with $x_N + \Delta x_N$ which maximize the increment of objective value. The following properties of l_a are important:

1. The general constraints are maintained along the direction l_a since
$$Ax(\theta) = Ax + \theta A(z - x) = (1 - \theta)Ax + \theta Az = b.$$

2. The objective value increases along l_a:
$$\frac{\partial c^T x}{l_a} = \frac{\partial c^T(x + \theta l_a)}{\partial \theta} = \frac{\partial F(x(\theta))}{\partial \theta} = c^T l_a = c^T z - c^T x = -\Delta_N^T l_N = \beta(x, J_{supp}) > \epsilon \geq 0.$$

Follow to 2) , the step length θ along l_a should be chosen without violating the simple constraints. This step is less then 1 because $x(1) = x + l_a = z$, but the case $d_* \leq z \leq d^*$ has not been realized.

For $0 \leq \theta < 1$ the non-support components do not violate the simple constraints since $d_{*N} \leq x_N \leq d^*_N$ and $d_{*N} \leq z_N \leq d^*_N$ in accordance with their determination. Consequently, increasing θ on the interval $[0, 1)$ can violate only the supporting simple constraints

$$d_{*supp} \leq x_{supp}(\theta) \leq d^*_{supp}. \tag{5.85}$$

Next find the maximal θ^o which guaranties the validity of constraints (5.85) which can be rewritten in the component form as

$$d_{*j} \leq x_j(\theta) = x_j + \theta l_{aj} \leq d^*_j, \quad j \in J_{supp}. \tag{5.86}$$

Denote by θ_j the maximal step length determined by $j-$th constraint of (5.86). For each j it is possible only three cases:

1. $l_j > 0$, the component $x_j(\theta)$ increases and achieves the critical value d^*_j with $\theta = \theta_j = \frac{d^*_j - x_j}{l_{aj}}$;

2. $l_j < 0$, the function $x_j(\theta)$ decreases and achieves the critical value d_{*j} with $\theta = \theta_j = \frac{d_{*j} - x_j}{l_{aj}}$;

3. $l_j = 0$, the component $x_j(\theta)$ does not change $x_j(\theta) = x_j$, i.e. we can put $\theta_j = \infty$.

Thus we have the following formula for the step length

$$\theta_j = \begin{cases} \frac{d^*_j - x_j}{l_{aj}}, & for\ l_j > 0, \\ \frac{d_{*j} - x_j}{l_{aj}}, & for\ l_j < 0, \\ \infty, & for\ l_j = 0. \end{cases} \tag{5.87}$$

The maximal step length θ^o with respect to the components $x_j(\theta),\ j \in J_{supp}$ is equal to

$$\theta^o = \theta_{j_o} = \min \theta_j,\ j \in J_{supp}. \qquad (5.88)$$

The index $j_o \in J_{supp}$ indicates the first component $x_{j_o}(\theta^o)$ that reaches the bound of simple constraints when θ is increasing. If $\{x, J_{supp}\}$ is not degenerate SF-point then $\theta^o > 0$ since $\theta_j > 0,\ \forall\ j \in J_{supp}$.
Thus, the desired feasible point $\bar{x} = x(\theta^o) = x + \theta^o l_a$ has been constructed. Also the new suboptimality estimate for the new SF-point $\{\bar{x}, J_{supp}\}$ is

$$\beta(\bar{x}, J_{supp}) = \Delta_N^T(\bar{x}_N - z_N) = \Delta_N^T(x_N + \theta^o l_N - z_N) =$$
$$= \Delta_N^T(1-\theta^o)(x_N - z_N) = (1-\theta^o)\beta(x, J_{supp}) \leq \beta(x, J_{supp}).$$

It is clear that the transformation $x \to \bar{x}$ satisfies the basic principle: the suboptimality estimate does not increase, and in the case of non-degeneracy, strictly decreases.
If

$$\beta(\bar{x}, J_{supp}) = (1-\theta^o)\beta(x, J_{supp}) \leq \epsilon.$$

then the solution process is stopped since the obtained vector $\bar{x}$ is an ϵ-optimal feasible point. Otherwise we go to the second part of the iteration procedure.

5.3.2 Procedure for the changing of support $J_{supp} \to \bar{J}_{supp}$

After the first procedure we have been constructed the new feasible solution $\bar{x}$ and suboptimality estimate $\beta(\bar{x}, J_{supp}) > \epsilon$.

Remember, that suboptimality estimates $\beta(\bar{x}, J_{supp})$ can be represented as the decomposition (5.72). Since, this property allows us to reduce the measure of non-optimality further. Namely, we will continue with the principle of decreasing the degree of non-optimality of the support J_{supp}: $\beta(J_{supp}) \geq \beta(\bar{J}_{supp})$.

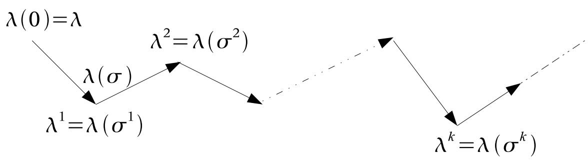
It was shown that degree of non-optimality of the support J_{supp} is a deviation of the dual objective function $\Phi(\lambda_{acc})$ from optimal one $\Phi(\lambda^0)$: $\beta(J_{supp}) = \Phi(\lambda_{acc}) - \Phi(\lambda^o)$. Also it was shown, that every support J_{supp} determines the accompanying dual feasible point λ_{acc}.

Thus in order to realize the principle of decreasing the degree of non-optimality of support, we need to construct the accompanying dual feasible point λ_{acc} using "old" support J_{supp}. Then change this accompanying dual feasible point by better one $\bar{\lambda}_{acc}$, such that $\Phi(\bar{\lambda}_{acc}) \leq \Phi(\lambda_{acc})$. The general schema of the second procedure of the primal adaptive method iteration is shown below:

$$\begin{array}{ccc} J_{supp} & & \bar{J}_{supp} \\ \Downarrow & & \Uparrow \\ \lambda_{acc} & \underset{\Phi(\bar{\lambda}_{acc})\leq\Phi(\lambda_{acc})}{\Longrightarrow} & \bar{\lambda}_{acc} \end{array} \tag{5.89}$$

The new dual accompanying feasible point $\bar{\lambda}$ will be calculate in a such way that $\beta(\bar{J}_{supp}) = \Phi(\bar{\lambda}) - \Phi(\lambda^o) \leq \Phi(\lambda) - \Phi(\lambda^o) = \beta(J_{supp})$.

To satisfying the requirements above we construct a continuous piecewise linear curve $\lambda(\sigma)$ in space of dual feasible points: $\lambda(\sigma) = (y(\sigma), v(\sigma), w(\sigma)), \sigma \geq 0$, see Figure 5.1(a), which started at $\lambda(0) = \lambda_{acc}$ and consist of the coordinated dual feasible points. The angle points of this curve are accompanying dual feasible points $\lambda^k = \lambda(\sigma^k), k = 1, 2, 3, ..., k_o$. This curve have the following property: along its first arc the dual cost function is decrease



(a) Curve $\lambda(\sigma)$

with a constant rate $\alpha^1, (\alpha^1 < 0)$ in non-degenerate case; further the descent rate is reduced by absolute value from arc to arc, i.e. $\alpha^{k+1} > \alpha^k$, where α^k is a descent rate of the dual objective function along the arc k.

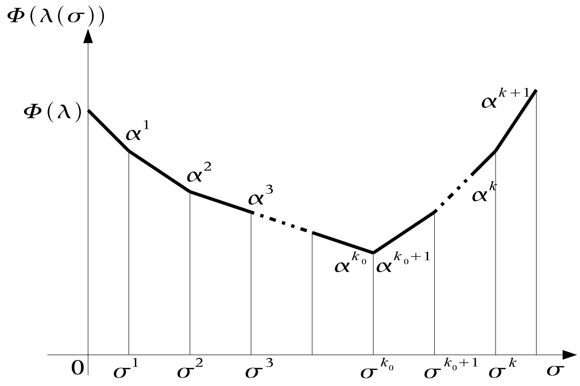
The transition $\lambda_{acc} \to \lambda_1 = \bar{\lambda}_{acc}$ is called "the short dual step rule".

And the substitution $\lambda_{acc} \longrightarrow \lambda^{k_o} = \bar{\lambda}_{acc}$ is called "the long dual step rule", if the point λ^{k_o} is a minimum of the dual objective cost function along the curve $\lambda(\sigma), \sigma \geq 0$ (i.e the angle point such that the value of the dual cost function decreases up to this point and then constat or increases (see Figure 5.1(b)).

To construct the curve $\lambda(\sigma), \sigma \geq 0$ construct the admissible direction $\Delta\lambda$ at the point λ_{acc} for the dual cost function $\Phi(\lambda), \lambda \in \Lambda$, where$\Lambda$ is a set of dual feasible points, i.e.

$$\frac{\partial\Phi(\lambda)}{\partial\Delta\lambda}|_{\lambda=\lambda_{acc}} = \lim_{\sigma\downarrow 0} \frac{\Phi(\lambda_{acc} + \sigma\Delta\lambda) - \Phi(\lambda_{acc})}{\sigma} < 0$$

After the first procedure the step $\theta^o = \theta_{jo}$.



(b)

Definition 8. *The SF-point $\{x, J_{supp}\}$ is called dually non-degenerate if their non-support components of the accompanying co-point are not equal to zero:*

$$\delta_j = \Delta_j \neq 0, \;\; j \in J_N. \tag{5.90}$$

The SF-point $\{x, J_{supp}\}$ satisfying to

$$d_{*j} < x_j < d^*_j, \quad j \in J_{supp} \tag{5.91}$$

is called primally non-degenerate.

The SF -point $\{x, J_{supp}\}$ will be called non-degenerate if it satisfies (5.90)-(5.91).

Determine a rule of variation of support components of co-vector

$$\delta_{supp}(\sigma) = \delta_{supp} + \sigma\Delta\delta_{supp}, \tag{5.92}$$

where

$$\Delta\delta_{supp} = -e_{j_o} sgn l_{j_o}, \;\; l_{j_o} = z_{j_o} - \bar{x}_{j_o} \tag{5.93}$$

and the unit vector $e_{j_0} \in \mathbb{R}^{|J_{supp}|}$ is given as $e_{j_0} = (\nu_j, \; j \in J_{supp}), \nu_j = 0, j \neq j_0, \; \nu_{j_0} = 1$. According to the (5.92)-(5.93), among the support components of the co-point we change only δ_j , corresponds to the component of the feasible point $\bar{x}_{j_o} = x_{j_o}(\theta^o)$ which comes first to the bound:

$$\bar{x}_{j_o} = \begin{cases} d_{*j_o}, & \text{if} \;\; l_{j_o} < 0 \\ d^*_{j_o}, & \text{if} \;\; l_{j_o} > 0. \end{cases} \tag{5.94}$$

The component δ_{j_o} is changed in such a way that its new value $\delta_{j_o}(\sigma)$ satisfy (5.44) for $\sigma \geq 0$ together with the component $\bar{x}_{j_o}$. Owing the (5.92)-(5.93) the last property is holds since

$$\begin{cases} \text{if } \ l_{j_o} < 0 \overset{\text{we have}}{\Longrightarrow} & \delta_{j_o}(\sigma) \geq \delta_{j_o} = 0, \quad \bar{x}_{j_o} = d_{*j_o}, \\ \text{if } \ l_{j_o} > 0 \overset{\text{we have}}{\Longrightarrow} & \delta_{j_o}(\sigma) \leq \delta_{j_o} = 0, \quad \bar{x}_{j_o} = d^*_{j_o}. \end{cases} \tag{5.95}$$

Then we need to determinate as a preliminary the changing direction $\Delta\delta_{supp}$ of the support component $\delta_j = \Delta_j, j \in J_{supp}$ of the vector of estimates :

$$\Delta\delta_j = 0, j \in J_{supp} \setminus j_o; \Delta\delta_{j_o} = \pm 1.$$

The choice of the sign will be realize in a such way that its new value $\delta_{j_o}(\sigma) = \sigma\Delta\delta_{j_o}$ satisfies (5.44) for small $\sigma \geq 0$ together with the component $\bar{x}_{j_o}$:

$$\Delta\delta_{j_o} = \begin{cases} +1, & \bar{x}_{j_o} = d_{*j_o}; \\ -1, & \bar{x}_{j_o} = d^*_{j_o}, \end{cases} = -sgn(l_{aj_o}) = -sgn(z_{j_o} - \bar{x}_{j_o}). \tag{5.96}$$

Thus we have been determined the rule of the transformation of the support components $\Delta\delta_{supp}$:

$$\Delta\delta_{supp} = -e_{j_o} sign\ l_{aj_o}, \quad l_{aj_o} = z_{j_o} - \bar{x}_{j_o}, \quad e_{j_o} \in \mathbf{R}^{|J_{supp}|} \text{ is the component of unit vector.} \tag{5.97}$$

Remind that the index j_o corresponds to the j_o- th coordinate which is critical in sense of (5.88).

Thus we fully defined the rule (5.92) for calculation of $\delta_{supp}(\sigma)$. Then for the given $\delta_{supp}(\sigma)$ and equation

$$\delta^T_{supp}(\sigma) = \underbrace{y^T(\sigma)}_{\overset{\shortparallel}{y+\sigma\Delta y}} A_{supp} - c^T_{supp}. \tag{5.98}$$

We find $y^T(\sigma)$:

$$\begin{aligned} y^T(\sigma) &= (\delta^T_{supp}(\sigma) + c^T_{supp})A^{-1}_{supp} = \\ &= (\delta^T_{supp} + c^T_{supp})A^{-1}_{supp} + \sigma\Delta\delta^T_{supp}A^{-1}_{supp} = \\ &= \underbrace{y^T}_{\overset{\shortparallel}{c^T A^{-1}_{supp}}} + \sigma \underbrace{\Delta y^T}_{\overset{\shortparallel}{\underbrace{\Delta\delta^T_{supp}A^{-1}_{supp}}_{\overset{\shortparallel}{-e^T_{jo} sign\ l_{a_{jo}} A^{-1}_{supp}}}}} \end{aligned} \tag{5.99}$$

Thus we found the variation of the Lagrange vector

$$\Delta y = \begin{pmatrix} \Delta y_{supp} \\ \Delta y_N \end{pmatrix} = \begin{pmatrix} -e_{j_o}^T sign\ l_{a_{j_o}} A_{supp}^{-1} \\ 0 \end{pmatrix} \tag{5.100}$$

which is the first component of the vector $\Delta\lambda = (\Delta y, \Delta v, \Delta w)$.

The knowledge of the $y^T(\sigma)$ allows us to find the non- supporting components of the co-vector $\delta_j(\sigma), j \in J_N$. Namely, definition we have

$$\delta_N^T(\sigma) = y^T(\sigma)A_N - c_N^T = y^T A_N - c_N^T + \sigma\Delta y^T A_N = \delta_N^T + \sigma\Delta\delta_N^T \tag{5.101}$$

where

$$\Delta\delta_N^T = \Delta y^T A_N = -e_{j_o}^T A_{supp}^{-1} A_N sign\ l_{a_{j_o}} = e_{j_o}^T A_{supp}^{-1} A_N \Delta\delta_{j_o}. \tag{5.102}$$

The other projections of the curve $\lambda(\sigma)$ are constructed in according to (5.67) and (5.92),(5.93),(5.101),(5.102).

For the dually non-degeneracy case of the SF point, the signs of $\delta_j(\sigma), j \in J_N,$ for small σ are coincide with the signs of $\delta_j = \Delta_j, \forall j \in J_N$. Then according to the coordinated conditions (5.61) construct the $v(\sigma), w(\sigma)$ as the following :

$$\begin{aligned} v_j(\sigma) = \overset{\delta_j+\sigma\Delta\delta_j}{\overset{\shortparallel}{\delta_j(\sigma)}}\ ,\ w_j(\sigma) = 0 \quad & \text{if} \quad \delta_j > 0; \\ v_j(\sigma) = 0,\ w_j(\sigma) = \underset{-\delta_j-\sigma\Delta\delta_j}{\underset{\shortparallel}{-\delta_j(\sigma)}} \quad & \text{if} \quad \delta_j < 0,\ j \in J_N. \\ v_j(\sigma) = w_j(\sigma) = 0 \quad & \text{if} \quad j \in J_{supp} \setminus j_o. \end{aligned} \tag{5.103}$$

and for j_o:

$$\begin{aligned} v_{j_o}(\sigma) = \delta_{j_o}(\sigma) = \sigma,\ w_{j_o}(\sigma) = 0 \quad & \text{if} \quad \Delta\delta_{j_o} = 1 (i.e\ l_{a_{j_o}} < 0); \\ v_{j_o}(\sigma) = 0,\ w_{j_o}(\sigma) = -\delta_{j_o}(\sigma) = -\sigma \quad & \text{if} \quad \Delta\delta_{j_o} = -1 (i.e\ l_{a_{j_o}} > 0). \end{aligned} \tag{5.104}$$

Since for small $\sigma > 0$ the vector $\lambda(\sigma) = \lambda_{acc} + \sigma\Delta\lambda$ are dual feasible point. It is easy to show that constructed direction $\Delta\lambda = (\Delta y, \Delta v, \Delta w)$ is proper direction and admissible, while $\lambda(\sigma)$ is dual feasible point.

Calculate along the function $\lambda(\sigma), \sigma \geq 0$, the value of dual cost function:

$$\begin{aligned} \Phi(\lambda(\sigma)) = b^T y(\sigma) - d_*^T v(\sigma) + d^{*T} w(\sigma) = \\ = (b^T y - d_*^T v + d^{*T} w) + \sigma(b^T\Delta y - d_*^T\Delta v + d^{*T}\Delta w) = \Phi(\lambda_{acc}) + \sigma\Delta\Phi. \end{aligned} \tag{5.105}$$

here Δv and Δw calculated as:

$$\Delta v_j = \begin{cases} 0, & j \in J_{supp} \setminus j_o; \\ \Delta\delta_{j_o}, & \text{if } \Delta\delta_{j_o} = 1, j = j_o; \\ 0, & \text{if } \Delta\delta_{j_o} = -1, j = j_o; \\ 0, & \text{if } \delta_j < 0; \\ \Delta\delta_j, & \text{if } \delta_j > 0, \quad j \in J_N; \end{cases} \tag{5.106}$$

$$\Delta w_j = \begin{cases} 0, & j \in J_{supp} \setminus j_o; \\ 0, & \text{if } \Delta\delta_{j_o} = 1, j = j_o; \\ -\Delta\delta_{j_o}, & \text{if } \Delta\delta_{j_o} = -1, j = j_o; \\ -\Delta\delta_j, & \text{if } \delta_j < 0 \\ 0, & \text{if } \delta_j > 0 \quad j \in J_N. \end{cases}$$

And the change direction $\Delta\Phi$ of the dual cost function is calculated with help (5.97)-(5.106) as follows:

$$\begin{aligned} &\Delta\Phi = \\ &= b^T\Delta y - d_*^T\Delta v + d^{*T}\Delta w = b_{supp}^T\Delta y_{supp} - d_*^T\Delta v_N + d^{*T}\Delta w_N - d_*^T\Delta v_{j_o} + d^{*T}\Delta w_{j_o} = \\ &= \sum_{i\in I_{supp}} b_i^T\Delta y_i - \sum_{\substack{j \in J_N \\ \delta_j > 0}} z_j(\Delta\delta_j) + \sum_{\substack{j \in J_N \\ \delta_j < 0}} z_j(-\Delta\delta_j) + \begin{cases} \bar{x}_{j_o}(\Delta\delta_{j_o}), & l_{j_o} < 0 \to (\Delta\delta_{j_o} = +1) \\ -\bar{x}_{j_o}(-\Delta\delta_{j_o}), & l_{j_o} > 0 \to (\Delta\delta_{j_o} = -1) \end{cases} \\ &= \Delta y_{supp}^T b_{supp} - \Delta\delta_N^T z_N - \Delta\delta_{j_o}\bar{x}_{j_o} = \Delta y_{supp}^T A(I_{supp}, J)z - \Delta y_{supp}^T A(I_{supp}, J_N)z_N - \Delta\delta_{j_o}\bar{x}_{j_o} = \\ &= \Delta y_{supp}^T A_{supp} z_{supp} - \Delta\delta_{j_o}\bar{x}_{j_o} = \\ &= -e_{j_o}^T A_{supp}^{-1} A_{supp} z_{supp}\Delta\delta_{j_o} + \bar{x}_{j_o}\Delta\delta_{j_o} = -(z_{j_o} - \bar{x}_{j_o})sgn(z_{j_o} - \bar{x}_{j_o}) = \\ &= -|z_{j_o} - \bar{x}_{j_o}| = \alpha(j_o). \end{aligned}$$

Thus,

$$\frac{\partial\Phi(\lambda)}{\partial\Delta\lambda}|_{\lambda=\lambda_{acc}} = -|z_{j_o} - \bar{x}_{j_o}| = \alpha(j_o) < 0 \tag{5.107}$$

i.e. the vector $\Delta\lambda$ - the descent direction of the function $\Phi(\lambda), \lambda \in \Lambda$ at the point λ_{acc}, and , being at the same time the admissible direction it is the proper direction too.

Remark 10. *If the $\{x, J_{supp}\}$ is a dual non-degenerate feasible SF point, then for the case $\theta = \theta_{j_o}$ we have (5.107).*

Thus on the initial arc of the curve $\lambda(\sigma),\ \sigma \geq 0$, the dual objective function is linear with respect to σ and decreases at a constant rate:

$$\alpha = -|z_{j_o} - \bar{x}_{j_o}| < 0. \tag{5.108}$$

According to the above calculations a linear rule of the variation of function $\Phi(\lambda(\sigma)), \sigma \geq 0$ remains valid until the first zero of the non-support components $\delta_j(\sigma), j \in J_N$ appears. The value σ^1 determined by this zero can easily be calculated from (5.101):

$$\sigma^1 = \sigma_{j_1} = \min_{j \in J_N} \sigma_j, \tag{5.109}$$

$$\sigma_j = \begin{cases} \frac{-\delta_j}{\Delta\delta_j}, & \text{if } \delta_j \Delta\delta_j < 0; \\ \infty, & \text{otherwise.} \end{cases}$$

Set

$$\sigma^* = \sigma^1 = \sigma_{j_1},\ \bar{\lambda} = \lambda(\sigma^*),\ \bar{J}_{supp} = (J_{supp} \setminus j_o) \cup j_1.$$

By construction we have

$$\Phi(\bar{\lambda}) = \Phi(\lambda) + \sigma^* \alpha < \Phi(\lambda). \tag{5.110}$$

Show that $\bar{J}_{supp}$ is a support, and that $\bar{\lambda}$ is an accompanying dual feasible point.
Among the numbers $\sigma_j,\ j \in J_N$ (5.109) one can always find finite ones, since otherwise $\Phi(\lambda(\sigma)) \to -\infty$ for $\sigma \to \infty$ but that is impossible, owing due to (5.60) and consistency of constraints of the primal problem.
According to (5.109) the finiteness of σ_j implies that $\Delta\delta_{j_1} \neq 0$. The matrix $\bar{A}_{supp} = A(I, \bar{J}_{supp})$ obtained from $A_{supp} = A(I, J_{supp})$ by exchanging the column a_{j_o} by a_{j_1} is non-singular Consequently $\bar{J}$ is a support.
By construction $\bar{\lambda}$ is a coordinated dual feasible point. For the co-point δ we get $\bar{\delta}_{supp} = \bar{\delta}(\bar{J}_{supp}) = 0$:

$$\bar{\delta}_{j_1} = 0,\ \bar{\delta}(\bar{J}_{supp} \setminus j_1) = \bar{\delta}(J_{supp} \setminus j_o) = 0$$

i.e. $\bar{\lambda}$ is a dual feasible point accompanying the support $\bar{J}_{supp}$. Thus the scheme (5.89) has been completely realized. The described procedure for $J_{supp} \to \bar{J}_{supp}$ is called short step rule. By (5.110) the degree of the support non-optimality decreases with the $|\alpha|\sigma^*$. This value is positive if SF point $\{x, J_{supp}\}$ is dually non-degenerate. And suboptimality estimate decreases by the same value

$$\beta(\bar{x}, \bar{J}_{supp}) = \beta(\bar{x}, J_{supp}) + \alpha\sigma^* < \beta(\bar{x}, J_{supp}).$$

The transition $J_{supp} \to \bar{J}_{supp}$ completes the second part of iteration. The method for solving initial problem (5.2), where the short step rule used, is called the adaptive method with the short step. Upon iteration of the method we have converse of suboptimality estimate:

$$\beta(\bar{x}, \bar{J}_{supp}) = (1 - \theta^o)\beta(x, J_{supp}) + \alpha\sigma^*.$$

It is strictly decreases if the SF point is not degenerate.

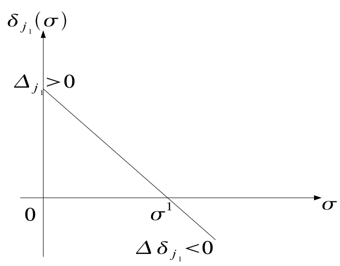
Let $\beta(\bar{x}, \bar{J}_{supp}) \leq \epsilon$. Then $\bar{x}$ is an ϵ-optimal feasible point. Otherwise we past to a new iteration with SF point $\{\bar{x}, \bar{J}_{supp}\}$.

As we said above the adaptive method have also the further possibilities for movements along $\lambda(\sigma), \sigma \geq 0$. The long step rule comes from the analysis of the behavior of the dual cost (5.105) while passing from one dual feasible solution to another. Namely it comes from the testing the behavior of the term $\zeta_j(\sigma), j \in J$ of the dual cost function:

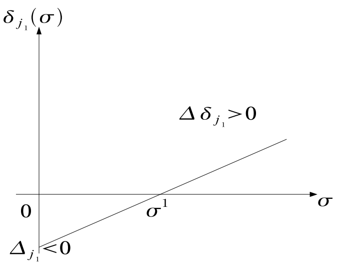
$$\Phi(\lambda(\sigma)) = \sum_{i \in I} b_i y_i(\sigma) + \sum_{j \in J} \underbrace{(-d_{*j} v_j(\sigma) + d_j^* w_j(\sigma))}_{\zeta_j(\sigma)} \tag{5.111}$$

Now explain this in more detail. Assume at the beginning that $\{x, J_{supp}\}$ is dually nondegenerate SF point, and at $\sigma = \sigma_1$ only the component of $\delta_N(\sigma)$ takes the zero value:

$$\delta_{j_1}(\sigma^1) = 0, \delta_j(\sigma^1) \neq 0, j \in J_N \setminus j_1.$$



(c) $\Delta_{j_1} > 0, \Delta\delta_{j_1} < 0$

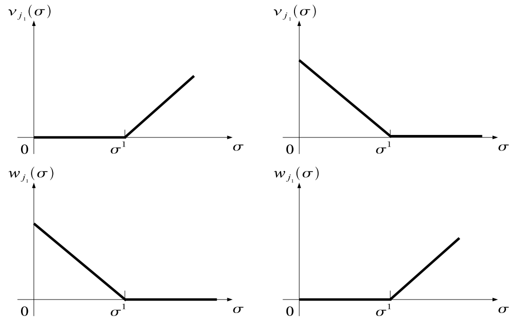


(d) $\Delta_{j_1} < 0, \Delta\delta_{j_1} > 0$

Then from the coordinated condition (5.61) follows that the linearity of the dual cost function(5.111) is violated at $\sigma = \sigma^1 = \sigma_{j_1}$ only for the components $v_{j_1}(\sigma), w_{j_1}(\sigma)$. Since, in the dual cost function the linear behavior of the expression $\zeta_{j_1}(\sigma)$ will be changed during the transition through the point σ_1.

At first case $(\Delta_{j_1} > 0, \Delta\delta_{j_1} < 0)$ we have:

$$\zeta_{j_1}(\sigma) = \begin{cases} d_{*j_1}(\Delta_{j_1} + \sigma\Delta\delta_{j_1}), & \sigma \leq \sigma^1; \\ -d_{*j_1}(-\Delta_{j_1} - \sigma\Delta\delta_{j_1}), & \sigma > \sigma^1. \end{cases} \tag{5.112}$$



(e) $\Delta_{j_1} > 0, \Delta\delta_{j_1} < 0$ (f) $\Delta_{j_1} < 0, \Delta\delta_{j_1} > 0$

Hence the rate of the function $\zeta_{j1}(\sigma), \sigma \geq 0$ at the point $\sigma = \sigma_1$ is changed on the following value

$$\Delta\alpha^{\zeta}_{j_1} = (d^*_{j_1} - d_{*j_1})(-\Delta\delta_{j_1}) = (d^*_{j_1} - d_{*j_1})|\Delta\delta_{j_1}| > 0 \tag{5.113}$$

and hence of the dual cost function is changed too.

By analogue we can show that for the second case $(\Delta_{j_1} < 0, \Delta\delta_{j_1} > 0)$ we have the same rate of change:

$$\Delta\alpha^{\zeta}_{j_1} = (d^*_{j_1} - d_{*j_1})|\Delta\delta_{j_1}| > 0 \tag{5.114}$$

Now assume that several components of the function $\delta_j(\sigma), \sigma \geq 0, j \in J^1 = \{j \in J_N : \delta_j(\sigma_1) = 0\}$ takes the zero value at the σ_1. Then the rate of change of the dual cost function is

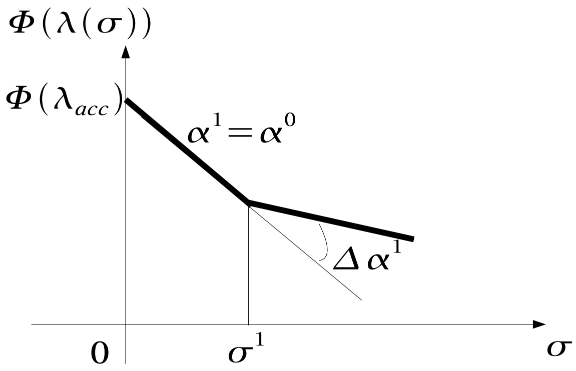
$$\Delta\alpha_{J^1} = \sum_{j \in J^1} (d^*_j - d_{*j})|\Delta\delta_j| > 0 \tag{5.115}$$

Let $\sigma^{k-1}, \sigma^k, 0 < \sigma^{k-1} < \sigma^k$ be a two arbitrary neighbouring zeros of the function $\delta_j(\sigma), \sigma \geq 0, j \in J^K = \{j \in J_N : \delta_j(\sigma^k) = 0\}$ and α_{k-1}be the rate of change of the dual cost function on $\sigma^{k-1} \leq \sigma \leq \sigma^k$. Then we get the following rule of the rate variation:

$$\Delta\alpha_k = \sum_{j \in J^K} (d^*_j - d_{*j})|\Delta\delta_j| \tag{5.116}$$

$$\alpha_k = \alpha_{k-1} + \Delta\alpha_k \qquad \text{on} \qquad \sigma^k \leq \sigma \leq \sigma^{k+1}.$$

Thus we can find the minimum value of the function $\Phi(\lambda(\sigma)), \sigma \geq 0$ along $\lambda(\sigma), \sigma \geq 0$ using the rule above. Namely, the function $\Phi(\lambda(\sigma)), \sigma \geq 0$ achieves the minimum value at



(g)

$\sigma = \sigma^* = \sigma^{k_o}$,

$$\alpha^{k_o-1} < 0, \alpha^{k_o} \geq 0. \tag{5.117}$$

The index k_o always exist, since $\inf \Phi(\lambda(\sigma)) > -\infty$

Set

$$\bar{\lambda} = \lambda(\sigma^*) \tag{5.118}$$

By construction we have

$$\Phi(\bar{\lambda}) = \Phi(\lambda) - \sum_{k=0}^{k_o-1} \alpha^k(\sigma^{k+1} - \sigma^k), \sigma^0 = 0. \tag{5.119}$$

Then we can construct a new support $\bar{J}_{supp}$ and show that $\bar{\lambda}$ is an accompanying dual feasible point. A new support will be constructed in the form:

$$\bar{J}_{supp} = (J_{supp} \setminus j_o) \cup j_*.$$

And to find a suitable index j_* we should consider two situations:1) $|J_{k_o}| = 1$; 2) $|J_{k_o} > 1|$. In the first case the set J_{k_o} consists of the only element which is taken as j_*.

Find j_* to be added to the support in the second case. Arrange the elements of the set J_{k_o} in the following way:

$$s_1, s_2, ..., s_q; \bigcup_{i=1}^{q} s_i = J_{k_o}; s_1 < s_2 < ... < s_q.$$

Then we can find such an element s_p for which:

$$\alpha_{s_{p-1}} = \alpha^{k_o+1} + \sum_{i=1}^{p-1}(d^*_{s_i} - d_{*s_i})|\Delta\delta_{s_i}| < 0, \quad (5.120)$$
$$\alpha_{s_p} = \alpha_{s_{p-1}} + (d^*_{s_p} - d_{*s_p})\Delta\delta_{s_p} \geq 0.$$

Such index as was shown above always exist. We set $j_* = s_p$. And the $\bar{J}_{supp}$ has been constructed.

This procedure called a long step rule. For $k_o = 1$ its automatically transformed into the short step rule. Since the degree of non-optimality of the support is decreased together with the dual objective function, then in accordance with (5.119)we will have the the new suboptimality estimate of the SF point:

$$\beta(\bar{x}, \bar{J}_{supp}) = (1 - \theta^o)\beta(x, J_{supp}) - \sum_{k=0}^{k_o-1} \alpha^k(\sigma^{k+1} - \sigma^k). \quad (5.121)$$

If $\beta(\bar{x}, \bar{J}_{supp} \leq \epsilon$ then $\bar{x}$ is an ϵ-optimal feasible point of the initial problem (5.2). Otherwise we start a new iteration with SF- point $\{\bar{x}, \bar{J}_{supp}\}$.

Remark 11. *In application of LP there is a situations when no priory information about feasible points available. In this case to use the primal adaptive method is not advisable. Assume that we have't any information about the feasible points of the problem (5.2). But we have an initial support J^1_{supp}. It can be shown that such information can be always provided [see ...]. Then we can start transform this initial support to obtain an optimal one: $J^1_{supp} \to J^2_{supp} \to \dots \to J^o_{supp}$. Then using the optimal support we construct the accompanying pseudo-feasible point z in accordance with (5.65), which will be an optimal feasible point $z = x^o$ in (5.2). To realize this idea we can use the second procedure of the primal adaptive method. As was shown above in order to start this procedure we will need the index j_o.*

To find this index we can do the following: by the support J^1_{supp} construct the pseudo feasible point $z^1_j, j \in J$. consider then only the support components of it $z^1_j, j \in J^1_{supp}$ and calculate the numbers

$\rho(z^1_j, [d_{*j}, d^*_j]), \quad j \in J^1_{supp}, \quad \rho(a, [b, c])$ *is a distance from the point* a *to interval* $[b, c]$,

and choose the maximal one among them: $\rho(z^1_{j_o}, [d_{*j_o}, d^*_{j_o}]) = \max_{j \in J^1_{supp}} \rho(z_j, [d_{*j}, d^*_j])$. *Then we can use the index j_o to change the support J^1_{supp} into J^2_{supp} with long or short step. The described process is called a dual adaptive method (with short or long step).*

Note that contrary to the simplex method of LP, where we iterate with feasible primal solution and infeasible dual solution that satisfy complementarity condition until we get dual feasibility, in adaptive method of LP we iterate with feasible primal and feasible dual solutions that do not satisfy complementarity condition until we get the complementarity condition fulfilled.

5.4 Algorithm

Assume that initial SF-point $\{x, J_{supp}\}$ and $\epsilon \geq 0$. are known.

Step 1: Calculate the multipliers and the support gradient:(5.36),(5.38)

$$u^T = c^T_{supp} A^{-1}_{supp}, \Delta^T = u^T A_{supp} - c^T,$$

and divide the set J_{supp} into two non-intersecting parts

$$J_N^+ = \{j \in J_N : \ \Delta_j \geq 0\}, \qquad J_N^- = \{j \in J_N : \ \Delta_j < 0\},$$
$$J_N^+ \cup J_N^- = J_N, \quad J_N^+ \cap J_N^- = \varnothing.$$

Determine by (5.74) the non-support components of the accompanying pseudo-feasible point z_N :

$$z_j = \begin{cases} d_{*j}, \ \text{for} j \in J_N^+ \\ d_j^*, \ \text{for} j \in J_N^- \end{cases}$$

Calculate the suboptimality estimate (5.75)

$$\beta(x, J_{supp}) = \sum_{j \in J_N} \Delta_j (x_j - z_j)$$

If $\beta(x, J_{supp}) \leq \epsilon$, then STOP, x is an ϵ-optimal feasible point. Otherwise go to Step 2.

Step 2 :

Calculate the support components of the pseudo- feasible point by (5.65):

$$z_{supp} = A^{-1}_{supp}(b - A_N z_N).$$

If $d_{*supp} \leq z_{supp} \leq d^*_{supp}$, then STOP, z is an optimal feasible point.

Otherwise determine by (5.84)

$$l_a = z - x$$

and calculate by (5.87)-(5.88) the step θ^o along l_a :

$$\theta^o = \theta_{j_o} = \min \theta_j,\ j \in J_{supp}$$

where

$$\theta_j = \begin{cases} \frac{d_j^* - x_j}{l_{a_j}}, & for\ l_{a_j} > 0, \\ \frac{d_{*j} - x_j}{l_{aj}}, & for\ l_{a_j} < 0, \\ \infty, & for\ l_{a_j} = 0, j \in J_{supp}. \end{cases}$$

Change the feasible point x :

$$\bar{x} = x + \theta^o l_a.$$

Calculate

$$\beta(\bar{x}, J_{supp}) = (1 - \theta^o)\beta(x, J_{supp})$$

If $\beta(\bar{x}, J_{supp}) \leq \epsilon$ then STOP : $\bar{x}$ is an ϵ-optimal feasible point. Otherwise go to Step 3.

Step 3:

Construct the direction $\Delta\delta_j, j \in J_N$ for changing the non-support component of the co-point $\delta = \Delta$ by (5.102):

$$\Delta\delta_N^T = -e_{j_o}^T A_{supp}^{-1} A_N \Delta\delta_{j_o}$$

where

$$\delta_{j_o} = \begin{cases} 1, & for\ \bar{x}_{j_o} = d_{*j_o}, \\ -1 & for\ \bar{x}_{j_o} = d_{j_o}^*. \end{cases}$$

For every $j \in J_N$ we calculate such σ_j that $\delta_j(\sigma) = \delta_j + \sigma\Delta\delta_j = 0$. We get

$$\sigma_j = \begin{cases} \frac{-\Delta_j}{\Delta\delta_j}, & \text{if } \Delta_j\Delta\delta_j < 0 \text{ or } j \in J_N^+,\ \Delta\delta_j < 0; \text{ or } j \in J_N^-, \Delta\delta_j > 0; \\ \infty, & \text{in other cases.} \end{cases}$$

Step 4:

Find j_* to be added to the support. Arrange the indexes $\{j \in J_N : \sigma_j \neq \infty\}$ in increasing values σ_j:

$$\sigma_{j_1} \leq \sigma_{j_2} \leq \dot{\leq} \sigma_{j_p}; j_k \in J_N,\ \sigma_{j_k} \neq \infty,\ k = \overline{1,p}.$$

For every $j_k,\ k = \overline{1,p}$ we calculate the jump of the rate of the dual objective function

$$\Delta\alpha_{j_k} = |\Delta\delta_{j_k}|(d^*_{j_k} - d_{*j_k}).$$

As j_* we choose j_q such that

$$\alpha_{j_{q-1}} = \alpha_o + \sum_{k=1}^{q-1} \Delta\alpha_{j_k} < 0, \quad \alpha_{j_q} = \alpha_o + \sum_{k=1}^{q} \Delta\alpha_{j_k} \geq 0$$

where $\alpha_o = -|z_{j_o} - \bar{x}_{j_o}|$ is the initial rate of dual objective function change.
For each $k = \overline{1, q-1}$ we set

$$\bar{z}_{j_k} = \begin{cases} d_{*j}, \text{ for } \ \Delta\delta_{j_*} > 0 \\ d^*_j, \text{ for } \ \Delta\delta_{j_*} < 0 \end{cases}$$

adding simultaneously j_k with $\Delta\delta_j > 0$ to $\bar{J}^+_N$ and j_k with $\Delta\delta_j < 0$ to $\bar{J}_N$.

We calculate

$$\Delta\Phi = \Phi(\lambda(\sigma_{j_q})) - \Phi(\lambda) = \sum_{k=1}^{q} \alpha_{j_{k-1}}(\sigma_{j_k} - \sigma_{j_{k-1}})$$

where $\alpha_{j_o} = \alpha_o$, $\sigma_{j_o} = \sigma_o = 0$. If

$$\beta(\bar{x}, \bar{J}_{supp}) = \beta(\bar{x}, J_{supp}) + \Delta\Phi \leq \epsilon$$

then STOP,$\bar{x}$ is an ϵ-optimal feasible point. Otherwise modify the support

$$J_{supp} \to \bar{J}_{supp} = (J_{supp} \setminus j_o) \cup j_*$$

and pass to a new iteration with SF-point $\{\bar{x}, \bar{J}_{supp}\}$ and the sets $\bar{J}^+_N, \bar{J}^-_N$.

Example

$$2x_1 + x_2 + 0 \cdot x_3 + ... + 0 \cdot x_6 \to \max$$

$$\begin{array}{rcrcrcrcrcrcr} x_1 & + & 2x_2 & - & x_3 & & & & & & & = & 2 \\ x_1 & + & 2x_2 & & & + & x_4 & & & & & = & 6 \\ -x_1 & + & x_2 & & & & & - & x_5 & & & = & -2 \\ -x_1 & + & x_2 & & & & & & & + & x_6 & = & 1 \end{array} \quad A(I,J) = \begin{pmatrix} 1 & 2 & -1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$1 \leq x_1 \leq 3,\ \ 0 \leq x_2 \leq 2,\ \ 0 \leq x_i \leq +\infty,\ i = \overline{3,6}$.

In accordance with our notation we have

$$\begin{aligned} I = \{1,2,3,4\}, \quad J = \{1,2,3,4,5,6\}, \\ x = (x_1, x_2, x_3, x_4, x_5, x_6) = x(J) \\ b(I) = (2, 6, -2, 1) \\ d^* = (3, 2, +\infty, +\infty, +\infty, +\infty) = d^*(J), \quad d_* = (1, 0, 0, 0, 0, 0) = d_*(J) \\ c = (2, 1, 0, 0, 0, 0) = c(J). \end{aligned}$$

1) Choose non-empty support $J^{(1)}_{supp} = \{3,4,5,6\}$, $J_N = \{1,2\}$, the corresponding support matrix is

$$A(I, J^{(1)}_{supp}) = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{since} \quad \det A(I, J^{(1)}_{supp}) = 1 \neq 0.$$

It easy to check that

$$A^{-1}(I, J^{(1)}_{supp}) = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Consider the arbitrary feasible point $x = (2, 1, 2, 2, 1, 2)$. Calculate the potentials

$$u = c^T_{supp} A^{-1}_{supp} = (0, 0, 0, 0) \cdot \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = (0, 0, 0, 0)$$

then $\Delta = -c_N \quad \Longrightarrow \Delta = (-2, -1, 0, 0, 0, 0)$. Clear, that optimality conditions are not satisfied. Divide the set J_N:

- $J^+_N = \emptyset$,
- $J^-_N = \{1, 2\}$

Then

$$\begin{cases} z_1 = 3, & \text{since } \{1\} \in J^-_N \\ z_2 = 2, & \text{since } \{2\} \in J^-_N \end{cases}$$

$$\beta(x, J^{(1)}_{supp}) = \Delta^T_N (x_N - z_N) = (-2, -1) \cdot \left[\begin{pmatrix} 2 \\ 1 \end{pmatrix} - \begin{pmatrix} 3 \\ 2 \end{pmatrix} \right] = (-2, -1) \cdot \begin{pmatrix} -1 \\ -1 \end{pmatrix} = 3$$

2) Find the new feasible point $\bar{x} = x + \theta^o \cdot l_a$, where

$$l_a = z - x, \text{ and } \quad z = \begin{cases} d_{*j}, & if \ \ \Delta_j > 0; \\ d_j^*, & if \ \ \Delta_j < 0, \ j \in J_N; \\ d_{*j} \text{ or } \ d_j^*, & if \ \ \Delta_j = 0. \end{cases} \quad \Longrightarrow$$

$$\begin{cases} z_1 = 3, & \text{since } \ \Delta_1 = -2 < 0 \\ z_2 = 2, & \text{since } \ \Delta_2 = -1 < 0 \end{cases}$$

$$\Downarrow$$

$$z_{supp} = A_{supp}^{-1}(b - A_N z_N) = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \left[\begin{pmatrix} 2 \\ 6 \\ -2 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 2 \\ 1 & 2 \\ -1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \end{pmatrix} \right] = \begin{pmatrix} 5 \\ -1 \\ 1 \\ 2 \end{pmatrix}$$

The supporting components of pseudo-feasible point is $z_{supp} = z_{(3,4,5,6)} = (5, -1, 1, 2)$. It is easy to check that its not satisfies to the prime constraints $d_{supp}^* \leq z_{supp} \leq d*_{supp}$ by supporting components, namely :

$$\begin{array}{llll} d_{*3} = 0 \leq & z_3 = 5 & \leq d_3^* = \infty & \checkmark \\ d_{*4} = 0 \leq & z_4 = -1 & \leq d_4^* = 2 & \maltese \\ d_{*5} = 0 \leq & z_5 = 1 & \leq d_5^* = \infty & \checkmark \\ d_{*6} = 0 \leq & z_6 = 2 & \leq d_6^* = \infty & \checkmark \end{array}$$

Then we need to continue with construction of $\bar{x}$.

Find admissible direction l_a:

$$l_a = z - x = \begin{pmatrix} 3 \\ 2 \\ 5 \\ -1 \\ 1 \\ 2 \end{pmatrix} - \begin{pmatrix} 2 \\ 1 \\ 2 \\ 2 \\ 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 3 \\ -3 \\ 0 \\ 0 \end{pmatrix}$$

And the the maximal step θ^o length among $\theta_j, j \in J_{supp} = \{3, 4, 5, 6\}$:

$$\theta_j = \begin{cases} \frac{d_j^* - x_j}{l_{aj}}, & for \ l_j > 0, \\ \frac{d_{*j} - x_j}{l_{aj}}, & for \ l_j < 0, \\ \infty, & for \ l_j = 0. \end{cases} \Longrightarrow \begin{cases} \theta_3 = \frac{d_3^* - x_3}{l_{a3}} = \frac{\infty - 2}{3} = \infty, & for \ l_3 = 3 > 0, \\ \theta_4 = \frac{d_{*4} - x_4}{l_{a4}} = \frac{0-2}{-3} = \frac{2}{3}, & for \ l_4 = -3 < 0, \\ \theta_5 = \theta_6 = \infty, & since \ l_5 = l_6 = 0. \end{cases}$$

The maximal step length θ^o with respect to the components $x_j(\theta),\ j \in J_{supp}$ is equal to

$$\theta^o = \theta_{j_o} = \min\{\theta_3, \theta_4, \theta_5, \theta_6\} = \min\{\infty, \frac{2}{3}, \infty, \infty\} = \frac{2}{3}$$

So we have $\theta^o = \theta_{j_o} = \theta_4$ and the index $j_o = \{4\}$.

Now we can calculate the new feasible point $\bar{x} = x + \theta^o l_a = \begin{pmatrix} 2 \\ 1 \\ 2 \\ 2 \\ 1 \\ 2 \end{pmatrix} + \frac{2}{3} \cdot \begin{pmatrix} 1 \\ 1 \\ 3 \\ -3 \\ 0 \\ 0 \end{pmatrix} =$

$\begin{pmatrix} \frac{8}{3} \\ \frac{5}{3} \\ 4 \\ 0 \\ 1 \\ 2 \end{pmatrix}$ Calculate the new estimate $\beta(\bar{x}, J_{supp})$:

$$\beta(\bar{x}, J_{supp}) = (1 - \theta^o)\beta(x, J_{supp}) = \frac{1}{3} \cdot 3 = 1$$

3) Construct the direction $\Delta\delta_N$ for changing the non support component of the co-vector

$$\delta = \begin{pmatrix} -2 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \Delta :$$

$$\Delta\delta_N^T(\sigma) = \Delta y_{supp}^T A_N = -e_{j_o}^T A_{supp}^{-1} A_N sign(l_{a_{j_o}}) = e_{j_o}^T A_{supp}^{-1} A_N \Delta\delta_{j_o}$$

In our case we have

$$\Delta\delta_{j_o} = \begin{cases} +1, & \bar{x}_{j_o} = d_{*j_o}; \\ -1, & \bar{x}_{j_o} = d^*_{j_o}. \end{cases} \overset{\text{or the sign can be defined by:}}{\Longleftrightarrow} \Delta\delta_{j_o} = -sgn(l_{a_{j_o}}) = -sgn(z_{j_o} - \bar{x}_{j_o})$$

$$\Delta\delta_4 = \left\{ +1, \quad \bar{x}_4 = 0. \right. \overset{\text{or alternatively we have the same sign}}{\Longleftrightarrow} \Delta\delta_4 = -sgn(l_4) = -sgn(\underset{-1}{z_4} - \underset{0}{\bar{x}_4}) = +1$$

then

$$\Delta\delta_N^T(\sigma) = \Delta y_{supp}^T A_N = e_{j_o}^T A_{supp}^{-1} A_N \Delta\delta_4 \quad = \quad (0,1,0,0) \cdot \begin{pmatrix} 1 & 2 \\ 1 & 2 \\ -1 & 1 \\ -1 & 1 \end{pmatrix} \cdot (+1) = (1,2)$$

For every $j \in J_N = \{1,2\}$ we calculate such $\sigma = (\sigma_1, \sigma_2)$ that

$$\delta_j(\sigma) = \delta_j + \sigma\Delta\delta_j = 0, j \in J_N = \{1,2\}$$

In accordance with

$$\sigma_j = \begin{cases} \frac{-\delta_j}{\Delta\delta_j}, & \text{if } \delta_j\Delta\delta_j < 0 \text{ or } j \in J_N^+,\ \Delta\delta_j < 0; \text{ or } j \in J_N^-, \Delta\delta_j > 0; \\ \infty, & \text{in other cases.} \end{cases}$$

We get

$$\begin{cases} \delta_1\Delta\delta_1 = (-2) \cdot 1 = -2 < 0 \quad \text{then} \quad \sigma_1 = \frac{-\delta_1}{\Delta\delta_1} = \frac{-(-2)}{1} = 2, \\ \delta_2\Delta\delta_2 = (-1) \cdot 2 = -2 < 0 \quad \text{then} \quad \sigma_2 = \frac{-\delta_2}{\Delta\delta_2} = \frac{-(-1)}{2} = \frac{1}{2}, \end{cases}$$

Thus we have $\sigma_1 = 2,\ \sigma_2 = \frac{1}{2}$.

4) Find j_* to be added to the support $J_{supp} = \{3,4,5,6\}$. Arrange the indexes $\{j = \{1,2\} \in J_N : \sigma_j \neq \infty\}$ in increasing values σ_j:

$$\begin{array}{rclll} \sigma_{j_1} & \leq & \sigma_{j_2}, & j_k \in J_N, \quad \sigma_{j_k} \neq \infty, & k = 1,2 \\ \sigma_2 & < & \sigma_1, & j_1 = \{2\}, j_2 = \{1\} \in J_N. & \end{array}$$

For every $j_k,\ k = 1,2$ we calculate the jump of the rate of the dual objective function

$$\begin{aligned} \Delta\alpha_{j_1} &= |\Delta\delta_{j_1}|(d_{j_1}^* - d_{*j_1}) \\ &\Downarrow \\ \Delta\alpha_2 &= |\Delta\delta_2|(d_2^* - d_{*2}) = |2|(2-0) = 4 \end{aligned}$$

$$\begin{aligned} \Delta\alpha_{j_2} &= |\Delta\delta_{j_2}|(d_{j_2}^* - d_{*j_2}) \\ &\Downarrow \\ \Delta\alpha_1 &= |\Delta\delta_1|(d_1^* - d_{*1}) = |1|(3-1) = 2 \end{aligned}$$

As j_* we choose j_q such that

$$\alpha_{j_{q-1}} = \alpha_o + \sum_{k=1}^{q-1} \Delta\alpha_{j_k} < 0, \qquad \alpha_{j_q} = \alpha_o + \sum_{k=1}^{q} \Delta\alpha_{j_k} \geq 0$$

$$\Downarrow \qquad\qquad \Downarrow$$

$$\alpha_{j_1} = \alpha_o + \Delta\alpha_{j_1} = -1 + 4 = 3 > 0 \qquad \alpha_{j_2} = \alpha_o + \Delta\alpha_{j_2} = -1 + 2 = 1 > 0$$

where $\alpha_o = -|z_{j_o} - \bar{x}_{j_o}| = -|-1-0| = -1$ is the initial rate of dual objective function change.

Since $\alpha_o = \alpha_{j_o} < 0$ and $\alpha_{j_1} > 0$
then the index $j^* = j_q$ which should be added to the support is $j^* = j_1 = \{2\}$.

For each $k = \overline{1, q-1}$ we set (in our case k=o)

$$\bar{z}_{j_k} = \begin{cases} d_{*j}, \text{ for } \Delta\delta_{j_*} > 0 \implies \\ d_j^*, \text{ for } \Delta\delta_{j_*} < 0 \end{cases} \qquad \bar{z}_{j_o} = \bar{z}_4 = \left\{ d_{*4} = 0, \text{ since } \Delta\delta_{j_1} = \Delta\delta_2 = 2 > 0 \right.$$

adding simultaneously j_k with $\Delta\delta_j > 0$ to $\bar{J}_N^+$
and j_k with $\Delta\delta_j < 0$ to $\bar{J}_N$. $\qquad j_o = \{4\} \overset{adding}{\rightarrow} \bar{J}_N^+$
Thus we obtain the new vector

$$z = \begin{pmatrix} 3 \\ 2 \\ 5 \\ \boldsymbol{-1} \\ 1 \\ 2 \end{pmatrix} \implies \bar{z} = \begin{pmatrix} 3 \\ 2 \\ 5 \\ \boldsymbol{0} \\ 1 \\ 2 \end{pmatrix}$$

and the new set:

- $\bar{J}_N^+ = \{4\}$,
- $\bar{J}_N^- = \{1\}$.

We calculate

$$\Delta\Phi = \Phi(\lambda(\sigma_{j_q})) - \Phi(\lambda) = \sum_{k=1}^{q} \alpha_{j_{k-1}}(\sigma_{j_k} - \sigma_{j_{k-1}})$$

$$\Downarrow$$

$$\Delta\Phi = \Phi(\lambda(\sigma_{j_1})) - \Phi(\lambda) = \alpha_{j_o}(\sigma_{j_1} - \sigma_{j_o}) = \alpha_o\sigma_{j_1} = -1 \cdot \frac{1}{2} = \frac{1}{2}$$

where $\alpha_{j_o} = \alpha_o = (-1)$, $\qquad \sigma_{j_o} = \sigma_o = 0$.

And the new suboptimality estimate is

$$\beta(\bar{x}, \bar{J}_{supp}) = \beta(\bar{x}, J_{supp}) + \Delta\Phi = 1 + (-\frac{1}{2}) = \frac{1}{2} \leq \epsilon$$

then STOP,$\bar{x}$ is an ϵ-optimal feasible point.

Otherwise modify the support

$$\begin{array}{rcl} J_{supp} \to \bar{J}_{supp} & = & (J_{supp} \setminus j_o) \cup j_* \\ & \Downarrow & \\ J_{supp} \to \bar{J}_{supp} & = & (J_{supp} \setminus 4) \cup 2 \end{array}$$

and pass to a new iteration with SF-point $\{\bar{x}, \bar{J}_{supp}\}$ and the sets

- $\bar{J}_N^+ = \{4\}$,
- $\bar{J}_N^- = \{1\}$.

***Iteration 2*:**

1) Start the second iteration with the support $J_{supp}^{(2)} = \{3, 2, 5, 6\}$, $J_N = \{1, 4\}$, the corresponding support matrix is

$$A(I, J_{supp}^{(2)}) = \begin{pmatrix} -1 & 2 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad \text{since} \quad \det A(I, J_{supp}^{(2)}) \neq 0.$$

It easy to check that

$$A^{-1}(I, J_{supp}^{(2)}) = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & -1 & 0 \\ 0 & -\frac{1}{2} & 0 & 1 \end{pmatrix}$$

The new feasible point $x^{(2)} = (\frac{8}{3}, \frac{5}{3}, 4, 0, 1, 2)$. Calculate the potentials

$$u = c_{supp}^T A_{supp}^{-1} = (1, 0, 0, 0) \cdot \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & -1 & 0 \\ 0 & -\frac{1}{2} & 0 & 1 \end{pmatrix} = (-1, 1, 0, 0)$$

then $\Delta = u^T A_N - c_N^T = (-1, 1, 0, 0) \cdot \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ -1 & 0 \\ -1 & 0 \end{pmatrix} - (2, 0) = (0, 1) - (2, 0) \Longrightarrow \Delta^T =$ $(-2, 0, 0, 1, 0, 0)$. Clear, that optimality conditions are not satisfied:

$$\Delta_1 = -2 < 0 \quad \text{for} \quad (x_1^{(2)} = \frac{8}{3}) \neq (d_1^* = 3), \maltese$$
$$\Delta_4 = 1 > 0 \quad \text{for} \quad (x_4^{(2)} = 0) = (d_{*4} = 0), \checkmark \quad \{1, 4\} \in J_N$$

2) Find the new feasible point $\bar{x}^{(2)} = x^{(2)} + \theta^o \cdot l_a$, where

$$l_a = z - x, \text{ and } \quad z = \begin{cases} d_{*j}, & if \;\; \Delta_j > 0; \\ d_j^*, & if \;\; \Delta_j < 0, \; j \in J_N; \\ d_{*j} \text{ or } \; d_j^*, & if \;\; \Delta_j = 0. \end{cases} \quad \Longrightarrow$$

$$\begin{cases} z_1 = 3, & \text{since} \;\; \Delta_1 = -2 < 0 \\ z_4 = 0, & \text{since} \;\; \Delta_4 = 1 > 0 \end{cases}$$

$$\Downarrow$$

$$z_{supp} = A_{supp}^{-1}(b - A_N z_N) = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & -1 & 0 \\ 0 & -\frac{1}{2} & 0 & 1 \end{pmatrix} \cdot \left[\begin{pmatrix} 2 \\ 6 \\ -2 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ -1 & 0 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} 3 \\ 0 \end{pmatrix} \right] = \begin{pmatrix} 4 \\ 3/2 \\ 1/2 \\ 5/2 \end{pmatrix}$$

Then the supporting components of the pseudo-feasible point $z = (z_{supp}, z_N)$ is

$$z_{supp} = (z_3 = 4, \quad z_2 = \frac{3}{2}, \quad z_5 = \frac{1}{2}, \quad z_6 = \frac{5}{2}).$$

The conditions above also satisfies automatically for non-supporting components of of pseudo feasible point $z = (z_{supp}, z_N)$ by construction. Thus the constructed vector z is optimal one!

Thus we obtain the optimal solution

$$z = \begin{pmatrix} 3 \\ 3/2 \\ 4 \\ 0 \\ 1/2 \\ 5/2 \end{pmatrix} \qquad \Longrightarrow \qquad x^{\boldsymbol{optimal}} = \begin{pmatrix} 3 \\ 3/2 \\ 4 \\ 0 \\ 1/2 \\ 5/2 \end{pmatrix}$$

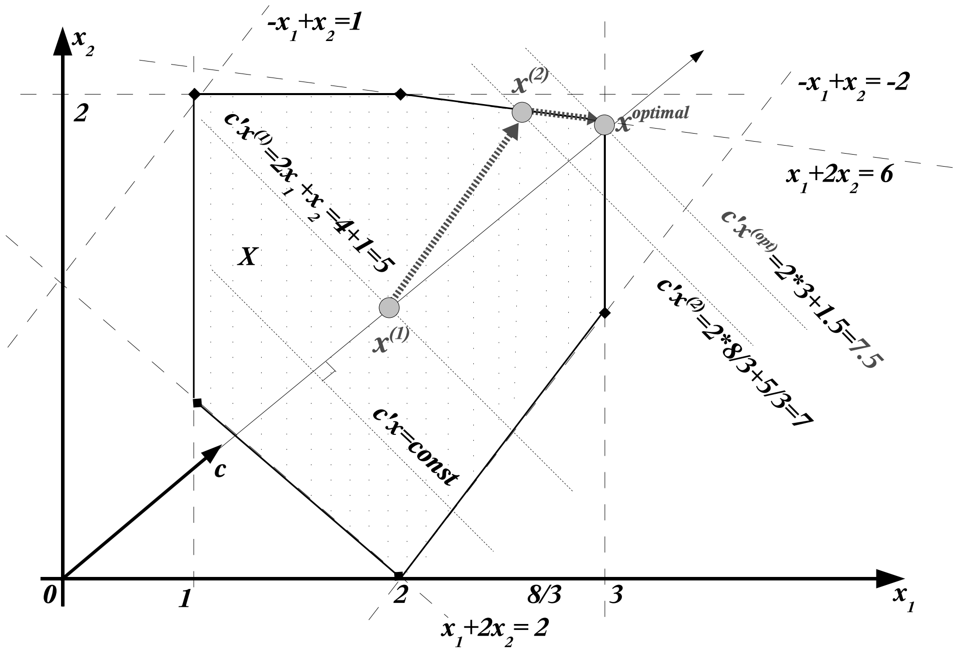
It is easy to check that constructed vector $x^{\boldsymbol{optimal}}$ satisfies to the conditions of the theorem 4:

$$\begin{cases} \Delta_j \geq 0 \;\; \text{for} \; x_j = d_{*j}, \\ \Delta_j \leq 0 \;\; \text{for} \; x_j = d_j^*, \\ \Delta_j = 0 \;\; \text{for} \;\; d_{*j} \leq x_j \leq d_j^*, \quad j \in J_N \end{cases}$$

The vector of estimates $\Delta^T = (-2, 0, 0, 1, 0, 0)$ then:

$$\begin{aligned}
\Delta_1 = -2 < 0 &\implies x_1 = 3 = d_3^* \quad \checkmark\\
\Delta_2 = 0 &\implies x_2 = 3/2 \in [0,2] \quad \checkmark\\
\Delta_3 = 0 &\implies x_3 = 4 \in [0,\infty) \quad \checkmark\\
\Delta_4 = 1 > 0 &\implies x_4 = 0 = d_{*4} \quad \checkmark\\
\Delta_5 = 0 &\implies x_5 = 1/2 \in [0,\infty) \quad \checkmark\\
\Delta_6 = 0 &\implies x_6 = 5/2 \in [0,\infty) \quad \checkmark
\end{aligned}$$

Thus we have been realized the principle of the decreasing the suboptimality estimates on



(h)

iteration of the method (see Fig 5.1(i)), namely:

$$\begin{aligned}
\beta(x^{(1)}, J^{(1)}_{supp}) &= 3, \quad (\text{ the exact value of deviation from optimum was } 7.5 - 5 = 2.5),\\
\beta(x^{(2)}, J^{(2)}_{supp}) &= \frac{1}{2}, \qquad (7.5 - 7 = 1/2, \text{too})
\end{aligned}$$

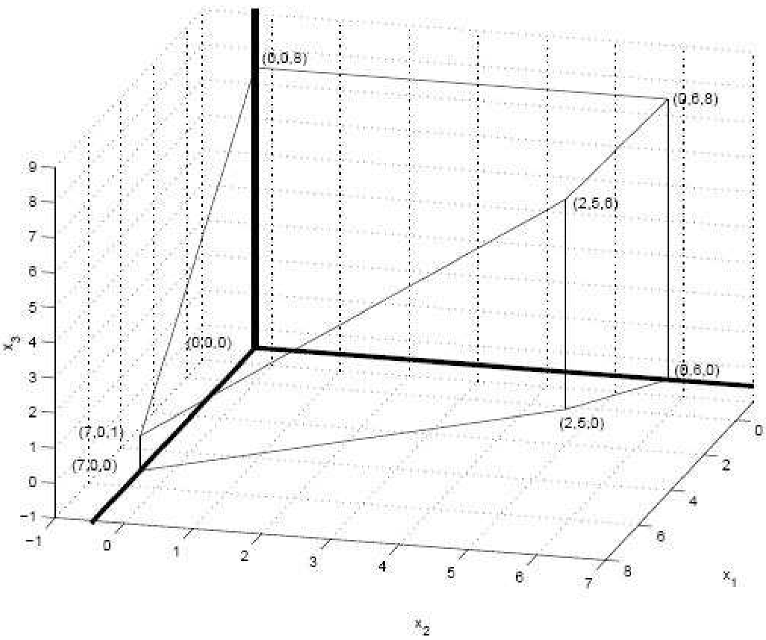
Finally we obtained $\beta(x^{opt}, J^{(2)}_{supp}) = 0$

Example 2 *(see also : Beispiel 5.7 "Optimierung I Einfuehrung in die Optimierung " von Prof. Dr. Alexander Martin, Prof. Dr. Mirjam Duer)*

$$
\begin{array}{llllllll}
 & 3x_1+2x_2+2x_3+0\cdot x_4...+0\cdot x_6\to \max & & & & & & \\
x_1 & & +\ x_3 & +x_4 & & & = & 8 \\
x_1 & +\ x_2 & & & +\ x_5 & & = & 7 \\
x_1 & +\ 2x_2 & & & & +\ x_6 & = & 12
\end{array}
\qquad
A(I,J)=\begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0\\ 1 & 1 & 0 & 0 & 1 & 0\\ 1 & 2 & 0 & 0 & 0 & 1 \end{pmatrix}
$$

$0\le x_i\le +\infty,\ i=\overline{1,6}.$
Note, that the constraints $x_i\ge 0,\ i=\overline{1,6}$ in the problem we modify as $0\le x_i\le M,\ i=\overline{1,6}$, where M means some a great number. Figure bellow shows we can put $M=8$.
In accordance with our notation we have

$$
\begin{aligned}
I=\{1,2,3\},\quad J=\{1,2,3,4,5,6\},\\
x=(x_1,x_2,x_3,x_4,x_5,x_6)=x(J)\\
b(I)=(8,7,12)\\
d^*=(+\infty,+\infty,+\infty,+\infty,+\infty,+\infty)=d^*(J),\quad d_*=(0,0,0,0,0,0,0)=d_*(J)\\
c=(3,2,2,0,0,0)=c(J).
\end{aligned}
$$



(i)

1) Choose non-empty support $J^{(1)}_{supp}=\{3,5,6\}$, $J_N=\{1,2,4\}$, the corresponding support matrix is

$$A(I, J^{(1)}_{supp}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{since} \quad \det A(I, J^{(1)}_{supp}) = 1 \neq 0.$$

It easy to check that

$$A^{-1}(I, J^{(1)}_{supp}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Consider the arbitrary feasible point $x = (2, 2, 2, 4, 3, 6)$. Calculate the potentials

$$u = c^T_{supp} A^{-1}_{supp} = (2, 0, 0) \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (2, 0, 0)$$

then $\Delta = uA_N - c_N \quad \Longrightarrow \Delta = (-1, -2, 2)$. Clear, that optimality conditions are not satisfied. Divide the set J_N:

- $J_N^+ = \{4\}$,
- $J_N^- = \{1, 2\}$

Then

$$\begin{cases} z_1 = 8, & \text{since } \{1\} \in J_N^- \\ z_2 = 8, & \text{since } \{2\} \in J_N^- \\ z_4 = 0, & \text{since } \{4\} \in J_N^+ \end{cases}$$

$$\beta(x, J^{(1)}_{supp}) = \Delta^T_N (x_N - z_N) = -1 \cdot (-6) + (-2)(-6) + 2 \cdot 4 = 26$$

2) Find the new feasible point $\bar{x} = x + \theta^o \cdot l_a$, where

$$l_a = z - x, \text{ and } \quad z = \begin{cases} d_{*j}, & if \;\; \Delta_j > 0; \\ d^*_j, & if \;\; \Delta_j < 0, \; j \in J_N; \\ d_{*j} \text{ or } d^*_j, & if \;\; \Delta_j = 0. \end{cases} \quad \Longrightarrow$$

$$\begin{cases} z_1 = 8, & \text{since } \; \Delta_1 = -1 < 0 \\ z_2 = 8, & \text{since } \; \Delta_2 = -2 < 0 \\ z_4 = 0, & \text{since } \; \Delta_2 = 2 > 0 \end{cases}$$

$$\Downarrow$$

$$z_{supp} = A^{-1}_{supp}(b - A_N z_N) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \left[\begin{pmatrix} 8 \\ 7 \\ 12 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 2 & 0 \end{pmatrix} \begin{pmatrix} 8 \\ 8 \\ 0 \end{pmatrix} \right] = \begin{pmatrix} 0 \\ -9 \\ -12 \end{pmatrix}$$

The supporting components of pseudo-feasible point is $z_{supp} = z_{(3,5,6)} = (0, -9, -12)$. It is easy to check that its not satisfies to the prime constraints $d^*_{supp} \leq z_{supp} \leq d*_{supp}$ by supporting components, namely :

$$\begin{array}{lccc}
d_{*3} = 0 \leq & z_3 = 0 & \leq d^*_3 = \infty & \checkmark \\
d_{*5} = 0 \leq & z_5 = -9 & \leq d^*_5 = \infty & \maltese \\
d_{*6} = 0 \leq & z_6 = -12 & \leq d^*_6 = \infty & \maltese
\end{array}$$

Then we need to continue with construction of $\bar{x}$.

Find admissible direction l_a:

$$l_a = z - x = \begin{pmatrix} 8 \\ 8 \\ 0 \\ 0 \\ -9 \\ -12 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \\ 2 \\ 4 \\ 3 \\ 6 \end{pmatrix} = \begin{pmatrix} 6 \\ 6 \\ -2 \\ -4 \\ -12 \\ -18 \end{pmatrix}$$

And the the maximal step θ^o length among $\theta_j, j \in J_{supp} = \{3, 5, 6\}$:

$$\theta_j = \begin{cases} \frac{d^*_j - x_j}{l_{aj}}, & for\ l_j > 0, \\ \frac{d_{*j} - x_j}{l_{aj}}, & for\ l_j < 0, \\ \infty, & for\ l_j = 0. \end{cases} \Longrightarrow \begin{cases} \theta_3 = \frac{d_{*3} - x_3}{l_{a3}} = \frac{0-2}{-2} = 1, & for\ l_3 = -2 < 0, \\ \theta_5 = \frac{d_{*5} - x_5}{l_{a3}} = \frac{0-3}{-12} = \frac{1}{4}, & for\ l_5 = -12 < 0, \\ \theta_6 = \frac{d_{*6} - x_6}{l_{a3}} = \frac{0-6}{-18} = \frac{1}{3}, & for\ l_6 = -18 < 0, \end{cases}$$

The maximal step length θ^o with respect to the components $x_j(\theta),\ j \in J_{supp}$ is equal to

$$\theta^o = \theta_{j_o} = \min\{\theta_3, \theta_5, \theta_6\} = \min\{1, \frac{1}{4}, \frac{1}{3}\} = \frac{1}{4}$$

So we have $\theta^o = \theta_{j_o} = \theta_5$ and the index $j_o = \{5\}$.

Now we can calculate the new feasible point $\bar{x} = x + \theta^o l_a = \begin{pmatrix} 2 \\ 2 \\ 2 \\ 4 \\ 3 \\ 6 \end{pmatrix} + \frac{1}{4} \cdot \begin{pmatrix} 6 \\ 6 \\ -2 \\ -4 \\ -12 \\ -18 \end{pmatrix} =$

$$\begin{pmatrix} \frac{7}{2} \\ \frac{7}{2} \\ \frac{3}{2} \\ 3 \\ 0 \\ \frac{3}{2} \end{pmatrix}$$ Calculate the new estimate $\beta(\bar{x}, J_{supp})$:

$$\beta(\bar{x}, J_{supp}) = (1-\theta^o)\beta(x, J_{supp}) = \frac{3}{4} \cdot 26 = \frac{39}{2}$$

3) Construct the direction $\Delta\delta_N$ for changing the non support component of the co-vector

$$\delta = \begin{pmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 0 \\ 0 \end{pmatrix} = \Delta :$$

$$\Delta\delta_N^T(\sigma) = \Delta y_{supp}^T A_N = -e_{j_o}^T A_{supp}^{-1} A_N sign(l_{a_{j_o}}) = e_{j_o}^T A_{supp}^{-1} A_N \Delta\delta_{j_o}$$

In our case we have

$$\begin{aligned} \Delta\delta_{j_o} &= \begin{cases} +1, & \bar{x}_{j_o} = d_{*j_o}; \\ -1, & \bar{x}_{j_o} = d^*_{j_o}. \end{cases} \overset{\text{or the sign can be defined by:}}{\iff} \Delta\delta_{j_o} = -sgn(l_{a_{j_o}}) = -sgn(z_{j_o} - \bar{x}_{j_o}) \\ \Delta\delta_5 &= \left\{ +1, \quad \bar{x}_5 = 0. \right. \overset{\text{or alternatively we have the same sign}}{\iff} \Delta\delta_5 = -sgn(l_5) = -sgn(\underset{-9}{z_5} - \underset{0}{\bar{x}_5}) = +1 \end{aligned}$$

then

$$\Delta\delta_N^T(\sigma) = \Delta y_{supp}^T A_N = e_{j_o}^T A_{supp}^{-1} A_N \Delta\delta_5 \quad = \quad (0,1,0) \cdot \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 2 & 0 \end{pmatrix} \cdot (+1) = (1,1,0)$$

For every $j \in J_N = \{1,2,4\}$ we calculate such $\sigma = (\sigma_1, \sigma_2, \sigma_4)$ that

$$\delta_j(\sigma) = \delta_j + \sigma\Delta\delta_j = 0, j \in J_N = \{1,2,4\}$$

In accordance with

$$\sigma_j = \begin{cases} \frac{-\delta_j}{\Delta\delta_j}, & \text{if } \delta_j\Delta\delta_j < 0 \text{ or } j \in J_N^+, \ \Delta\delta_j < 0; \text{ or } j \in J_N^-, \Delta\delta_j > 0; \\ \infty, & \text{in other cases.} \end{cases}$$

We get

$$\begin{cases} \delta_1\Delta\delta_1 = (-1)\cdot 1 = -1 < 0 & \text{then} \quad \sigma_1 = \frac{-\delta_1}{\Delta\delta_1} = \frac{-(-1)}{1} = 1, \\ \delta_2\Delta\delta_2 = (-2)\cdot 1 = -2 < 0 & \text{then} \quad \sigma_2 = \frac{-\delta_2}{\Delta\delta_2} = \frac{-(-2)}{1} = 2, \\ \delta_4\Delta\delta_4 = 2\cdot 0 = 0 & \text{then} \quad \sigma_4 = \infty \end{cases}$$

Thus we have $\sigma_1 = 1,\ \sigma_2 = 2.$

4) Find j_* to be added to the support $J_{supp} = \{3,5,6\}$. Arrange the indexes $\{j = \{1,2,4\} \in J_N : \sigma_j \neq \infty\}$ in increasing values σ_j:

$$\begin{array}{ccccc} \sigma_{j_1} & \leq & \sigma_{j_2}, & j_k \in J_N, \quad \sigma_{j_k} \neq \infty, & k = 1,2 \\ \sigma_1 & < & \sigma_2, & j_1 = \{1\}, j_2 = \{2\} \in J_N. & \end{array}$$

For every $j_k,\ k = 1,2$ we calculate the jump of the rate of the dual objective function

$$\begin{aligned} \Delta\alpha_{j_1} &= |\Delta\delta_{j_1}|(d^*_{j_1} - d_{*j_1}) \\ &\Downarrow \\ \Delta\alpha_1 &= |\Delta\delta_1|(d^*_1 - d_{*1}) = |1|(8-0) = \infty(8) \end{aligned}$$

$$\begin{aligned} \Delta\alpha_{j_2} &= |\Delta\delta_{j_2}|(d^*_{j_2} - d_{*j_2}) \\ &\Downarrow \\ \Delta\alpha_2 &= |\Delta\delta_2|(d^*_2 - d_{*2}) = |1|(8-0) = \infty(8) \end{aligned}$$

As j_* we choose j_q such that

$$\begin{array}{llll} \alpha_{j_{q-1}} & = \alpha_o + \sum_{k=1}^{q-1} \Delta\alpha_{j_k} < 0, & \alpha_{j_q} & = \alpha_o + \sum_{k=1}^{q} \Delta\alpha_{j_k} \geq 0 \\ \Downarrow & & \Downarrow & \\ \alpha_{j_o} & = \alpha_o = -9 < 0 & \alpha_{j_1} & = \alpha_o + \Delta\alpha_{j_1} = -9 + \infty > 0 \end{array}$$

where $\alpha_o = -|z_{j_o} - \bar{x}_{j_o}| = -|-9-0| = -9$ is the initial rate of dual objective function change.

Since $\alpha_{j_0} < 0$ and $\alpha_{j_1} > 0$
then the index $j^* = j_q$ which should be added to the support is $j^* = j_1 = \{1\}$.

For each $k = \overline{1, q-1}$ we set (in our case k=o, i.e short step rule)

$$\bar{z}_{j_k} = \begin{cases} d_{*j}, \text{ for } \Delta\delta_{j_*} > 0 \\ d_j^*, \text{ for } \Delta\delta_{j_*} < 0 \end{cases} \Longrightarrow \qquad \bar{z}_{j_0} = \bar{z}_5 = \left\{ d_{*5} = 0, \text{ since } \Delta\delta_{j_0} = \Delta\delta_5 = 1 > 0 \right.$$

adding simultaneously j_k with $\Delta\delta_j > 0$ to $\bar{J}_N^+$
and j_k with $\Delta\delta_j < 0$ to $\bar{J}_N$. $\quad j_0 = \{5\} \stackrel{adding}{\rightarrow} \bar{J}_N^+$
Thus we obtain the new vector

$$z = \begin{pmatrix} 8 \\ 8 \\ 0 \\ 0 \\ \boldsymbol{-9} \\ -12 \end{pmatrix} \Longrightarrow \bar{z} = \begin{pmatrix} 8 \\ 8 \\ 0 \\ 0 \\ \boldsymbol{0} \\ -12 \end{pmatrix}$$

and the new set:

- $\bar{J}_N^+ = \{4, 5\}$,
- $\bar{J}_N^- = \{2\}$.

We calculate

$$\begin{aligned} \Delta\Phi &= \Phi(\lambda(\sigma_{j_q})) - \Phi(\lambda) = \sum_{k=1}^{q} \alpha_{j_{k-1}}(\sigma_{j_k} - \sigma_{j_{k-1}}) \\ &\Downarrow \\ \Delta\Phi &= \Phi(\lambda(\sigma_{j_1})) - \Phi(\lambda) = \alpha_{j_o}(\sigma_{j_1} - \sigma_{j_o}) = \alpha_o\sigma_{j_1} = \alpha_0\sigma_1 = -9 \cdot 1 = -9 \end{aligned}$$

where $\alpha_{j_o} = \alpha_o = (-9), \qquad \sigma_{j_o} = \sigma_o = 0.$

And the new suboptimality estimate is

$$\beta(\bar{x}, \bar{J}_{supp}) = \beta(\bar{x}, J_{supp}) + \Delta\Phi = \frac{39}{2} - 9 = \frac{21}{2} \leq \epsilon$$

then STOP,$\bar{x}$ is an ϵ-optimal feasible point.

Otherwise modify the support

$$\begin{aligned} J_{supp} \to \bar{J}_{supp} &= (J_{supp} \setminus j_o) \cup j_* \\ &\Downarrow \\ J_{supp} \to \bar{J}_{supp} &= (J_{supp} \setminus 5) \cup 1 \end{aligned}$$

and pass to a new iteration with SF-point $\{\bar{x}, \bar{J}_{supp}\}$ and the sets

- $\bar{J}_N^+ = \{4, 5\}$,
- $\bar{J}_N^- = \{2\}$.

Iteration 2:

1) Start the second iteration with the support $J^{(2)}_{supp} = \{3,1,6\}$, $J_N = \{5,2,4\}$, the corresponding support matrix is

$$A(I, J^{(2)}_{supp}) = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \quad \text{since} \quad \det A(I, J^{(2)}_{supp}) \neq 0.$$

It easy to check that

$$A^{-1}(I, J^{(2)}_{supp}) = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

The new feasible point $x^{(2)} = (\frac{7}{2}, \frac{7}{2}, \frac{3}{2}, 3, 0, \frac{3}{2})$. Calculate the potentials

$$u = c^T_{supp} A^{-1}_{supp} = (2,3,0) \cdot \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} = (2,1,0)$$

then $\Delta = u^T A_N - c^T_N = (2,1,0) \cdot \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 2 & 0 \end{pmatrix} - (0,2,0) = (1,-1,2) \Longrightarrow \Delta^T =$ $(0,-1,0,2,1,0)$. Clear, that optimality conditions are not satisfied:

$$\begin{aligned} \Delta_2 = -1 < 0 \quad & \text{for} \quad (x^{(2)}_1 = \frac{7}{2}) \neq (d^*_1 = 8), \maltese \\ \Delta_5 = 1 > 0 \quad & \text{for} \quad (x^{(2)}_5 = 0) = (d_{*5} = 0), \checkmark \\ \Delta_4 = 2 > 0 \quad & \text{for} \quad (x^{(2)}_4 = 3) \neq (d_{*4} = 0), \maltese \quad \{5,2,4\} \in J_N \end{aligned}$$

2) Find the new feasible point $\bar{x}^{(2)} = x^{(2)} + \theta^o \cdot l_a$, where

$$l_a = z - x, \text{ and } \quad z = \begin{cases} d_{*j}, & if \ \ \Delta_j > 0; \\ d^*_j, & if \ \ \Delta_j < 0, \ j \in J_N; \\ d_{*j} \text{ or } d^*_j, & if \ \ \Delta_j = 0. \end{cases} \quad \Longrightarrow$$

$$\begin{cases} z_5 = 0, & \text{since} \quad \Delta_5 = 2 > 0 \\ z_2 = 8, & \text{since} \quad \Delta_2 = -1 < 0 \\ z_4 = 0, & \text{since} \quad \Delta_4 = 1 > 0 \end{cases}$$

$$\Downarrow$$

$$z_{supp} = A^{-1}_{supp}(b - A_N z_N) = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \cdot \left[\begin{pmatrix} 8 \\ 7 \\ 12 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 2 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 8 \\ 0 \end{pmatrix} \right] = \begin{pmatrix} 9 \\ -1 \\ -3 \end{pmatrix}$$

Then the supporting components of the pseudo-feasible point $z = (z_{supp}, z_N)$ is

$$z_{supp} = (z_3 = 9, \quad z_1 = -1, \quad z_6 = -3.)$$

It is easy to check that its not satisfies to the prime constraints $d^*_{supp} \leq z_{supp} \leq d*_{supp}$ by supporting components, namely :

$$\begin{array}{lccr} d_{*3} = 0 \leq & z_3 = 9 & \leq d^*_3 = \infty & \checkmark \\ d_{*1} = 0 \leq & z_1 = -1 & \leq d^*_1 = \infty & \maltese \\ d_{*6} = 0 \leq & z_6 = -3 & \leq d^*_6 = \infty & \maltese \end{array}$$

Find admissible direction l_a:

$$l_a = z - x = \begin{pmatrix} -1 \\ 8 \\ 9 \\ 0 \\ 0 \\ -3 \end{pmatrix} - \begin{pmatrix} 7/2 \\ 7/2 \\ 3/2 \\ 3 \\ 0 \\ 3/2 \end{pmatrix} = \begin{pmatrix} -9/2 \\ 9/2 \\ 15/2 \\ -3 \\ 0 \\ -9/2 \end{pmatrix}$$

And the the maximal step θ^o length among $\theta_j, j \in J_{supp} = \{3, 1, 6\}$:

$$\theta_j = \begin{cases} \frac{d^*_j - x_j}{l_{aj}}, & for\ l_j > 0, \\ \frac{d_{*j} - x_j}{l_{aj}}, & for\ l_j < 0, \\ \infty, & for\ l_j = 0. \end{cases} \Longrightarrow \begin{cases} \theta_3 = \frac{d^*_3 - x_3}{l_{a3}} = \frac{8-3/2}{15/2} = 13/15, & for\ l_3 = 15/2 > 0, \\ \theta_1 = \frac{d_{*1} - x_1}{l_{a1}} = \frac{0-7/2}{(-9/2)} = \frac{7}{9}, & for\ l_1 = -9/2 < 0, \\ \theta_6 = \frac{d_{*6} - x_6}{l_{a6}} = \frac{0-(3/2)}{-9/2} = \frac{1}{3}, & for\ l_6 = -9/2 < 0, \end{cases}$$

The maximal step length θ^o with respect to the components $x_j(\theta),\ j \in J_{supp}$ is equal to

$$\theta^o = \theta_{j_o} = \min\{\theta_3, \theta_1, \theta_6\} = \min\{\frac{13}{15}, \frac{7}{9}, \frac{1}{3}\} = \frac{1}{3}$$

So we have $\theta^o = \theta_{j_o} = \theta_6$ and the index $j_o = \{6\}$.

Now we can calculate the new feasible point $\bar{x} = x + \theta^o l_a = \begin{pmatrix} 7/2 \\ 7/2 \\ 3/2 \\ 3 \\ 0 \\ 3/2 \end{pmatrix} + \frac{1}{3} \cdot \begin{pmatrix} -9/2 \\ 9/2 \\ 15/2 \\ -3 \\ 0 \\ -9/2 \end{pmatrix} =$

$$\begin{pmatrix} 2 \\ 5 \\ 4 \\ 2 \\ 0 \\ 0 \end{pmatrix}$$ Calculate the new estimate $\beta(\bar{x}, J_{supp})$:

$$\beta(\bar{x}, J_{supp}) = (1 - \theta^o)\beta(x, J_{supp}) = \frac{2}{3} \cdot \frac{21}{2} = 7$$

3) Construct the direction $\Delta\delta_N$ for changing the non support component of the co-vector

$$\delta = \begin{pmatrix} 0 \\ -1 \\ 0 \\ 2 \\ 1 \\ 0 \end{pmatrix} = \Delta :$$

$$\Delta\delta_N^T(\sigma) = \Delta y_{supp}^T A_N = -e_{j_o}^T A_{supp}^{-1} A_N sign(l_{a_{j_o}}) = e_{j_o}^T A_{supp}^{-1} A_N \Delta\delta_{j_o}$$

In our case we have

$$\begin{aligned} \Delta\delta_{j_o} &= \begin{cases} +1, & \bar{x}_{j_o} = d_{*j_o}; \\ -1, & \bar{x}_{j_o} = d^*_{j_o}. \end{cases} \overset{\text{or the sign can be defined by:}}{\Longleftrightarrow} \Delta\delta_{j_o} = -sgn(l_{a_{j_o}}) = -sgn(z_{j_o} - \bar{x}_{j_o}) \\ \Delta\delta_6 &= \left\{ +1, \quad \bar{x}_6 = 0. \right. \overset{\text{or alternatively we have the same sign}}{\Longleftrightarrow} \Delta\delta_6 = -sgn(l_6) = -sgn(\underset{-3}{z_6} - \underset{0}{\bar{x}_6}) = +1 \end{aligned}$$

then

$$\Delta\delta_N^T(\sigma) = \Delta y_{supp}^T A_N = e_{j_o}^T A_{supp}^{-1} A_N \Delta\delta_6 \quad = \quad (0, 0, 1) \cdot \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 2 & 0 \end{pmatrix} \cdot (+1) = (0, 2, 0)$$

For every $j \in J_N = \{5, 2, 4\}$ we calculate such $\sigma = (\sigma_5, \sigma_2, \sigma_4)$ that

$$\delta_j(\sigma) = \delta_j + \sigma\Delta\delta_j = 0, j \in J_N = \{5, 2, 4\}$$

In accordance with

$$\sigma_j = \begin{cases} \frac{-\delta_j}{\Delta\delta_j}, & \text{if } \ \delta_j\Delta\delta_j < 0 \ \text{ or } \ j \in J_N^+, \ \Delta\delta_j < 0; \ \text{ or } \ j \in J_N^-, \Delta\delta_j > 0; \\ \infty, & \text{in other cases.} \end{cases}$$

We get

$$\begin{cases} \delta_5\Delta\delta_5 = 1\cdot 0 = 0 & \text{then} \quad \sigma_5 = \infty \\ \delta_2\Delta\delta_2 = (-1)\cdot 2 = -2 < 0 & \text{then} \quad \sigma_2 = \frac{-\delta_2}{\Delta\delta_2} = \frac{-(-1)}{2} = 1/2, \\ \delta_4\Delta\delta_4 = 2\cdot 0 = 0 & \text{then} \quad \sigma_4 = \infty \end{cases}$$

Thus we have $\sigma_5 = \infty, \sigma_2 = 1/2, \sigma_4 = \infty$.

4) Find j_* to be added to the support $J_{supp} = \{3,1,6\}$. Arrange the indexes $\{j = \{5,2,4\} \in J_N : \sigma_j \neq \infty\}$ in increasing values σ_j:

$$\begin{aligned} \sigma_{j_1} &\leq \sigma_{j_2}, \quad j_k \in J_N, \quad \sigma_{j_k} \neq \infty, \quad k = 1,2 \\ \sigma_{j_1} &= \sigma_2(\text{only one element, i.e northing to sort }), \qquad j_1 = \{2\} \in J_N. \end{aligned}$$

Calculate the jump of the rate of the dual objective function

$$\begin{aligned} \Delta\alpha_{j_1} &= |\Delta\delta_{j_1}|(d^*_{j_1} - d_{*j_1}) \\ &\Downarrow \\ \Delta\alpha_2 &= |\Delta\delta_2|(d^*_2 - d_{*2}) = |2|(8-0) = \infty(16) \end{aligned}$$

As j_* we choose j_q such that

$$\begin{aligned} \alpha_{j_{q-1}} &= \alpha_o + \sum_{k=1}^{q-1} \Delta\alpha_{j_k} < 0, & \alpha_{j_q} &= \alpha_o + \sum_{k=1}^{q} \Delta\alpha_{j_k} \geq 0 \\ &\Downarrow & &\Downarrow \\ \alpha_{j_o} &= \alpha_o = -3 < 0 & \alpha_{j_1} &= \alpha_o + \Delta\alpha_{j_1} = -3 + \infty > 0 \end{aligned}$$

where $\alpha_o = -|z_{j_o} - \bar{x}_{j_o}| = -|-3-0| = -3$ is the initial rate of dual objective function change.

Since $\alpha_{j_0} < 0$ and $\alpha_{j_1} > 0$
then the index $j^* = j_q$ which should be added to the support is $j^* = j_1 = \{2\}$.

For each $k = \overline{1, q-1}$ we set (in our case k=o)

$$\bar{z}_{j_k} = \begin{cases} d_{*j}, \text{ for } \Delta\delta_{j_*} > 0 \\ d^*_j, \text{ for } \Delta\delta_{j_*} < 0 \end{cases} \Longrightarrow \qquad \bar{z}_{j_0} = \bar{z}_6 = \left\{ d_{*6} = 0, \text{ since } \Delta\delta_{j_0} = \Delta\delta_6 = 1 > 0 \right.$$

adding simultaneously j_k with $\Delta\delta_j > 0$ to $\bar{J}_N^+$
and j_k with $\Delta\delta_j < 0$ to $\bar{J}_N$.

$$j_0 = \{6\} \overset{adding}{\to} \bar{J}_N^+$$

Thus we obtain the new vector

$$z = \begin{pmatrix} -1 \\ 8 \\ 9 \\ 0 \\ 0 \\ \boldsymbol{-3} \end{pmatrix} \quad \Longrightarrow \quad \bar{z} = \begin{pmatrix} -1 \\ 8 \\ 9 \\ 0 \\ 0 \\ \boldsymbol{0} \end{pmatrix}$$

and the new set:

- $\bar{J}_N^+ = \{4, 5, 6\}$,
- $\bar{J}_N^- = \{\emptyset\}$.

We calculate

$$\begin{aligned} \Delta\Phi &= \Phi(\lambda(\sigma_{j_q})) - \Phi(\lambda) = \sum_{k=1}^{q} \alpha_{j_{k-1}}(\sigma_{j_k} - \sigma_{j_{k-1}}) \\ &\Downarrow \\ \Delta\Phi &= \Phi(\lambda(\sigma_{j_1})) - \Phi(\lambda) = \alpha_{j_o}(\sigma_{j_1} - \sigma_{j_o}) = \alpha_o\sigma_{j_1} = \alpha_0\sigma_2 = -3 \cdot \frac{1}{2} = -\frac{1}{2} \end{aligned}$$

where $\alpha_{j_o} = \alpha_o = -3, \qquad \sigma_{j_o} = \sigma_o = 0.$

And the new suboptimality estimate is

$$\beta(\bar{x}, \bar{J}_{supp}) = \beta(\bar{x}, J_{supp}) + \Delta\Phi = 7 - \frac{3}{2} = \frac{11}{2} \leq \epsilon$$

then STOP,$\bar{x}$ is an ϵ-optimal feasible point.

Otherwise modify the support

$$\begin{aligned} J_{supp} \to \bar{J}_{supp} &= (J_{supp} \setminus j_o) \cup j_* \\ &\Downarrow \\ J_{supp} \to \bar{J}_{supp} &= (J_{supp} \setminus 6) \cup 2 \end{aligned}$$

and pass to a new iteration with SF-point $\{\bar{x}, \bar{J}_{supp}\}$ and the sets

- $\bar{J}_N^+ = \{4, 5, 6\}$,
- $\bar{J}_N^- = \{\emptyset\}$.

Iteration 3:

1) Start the third iteration with the support $J^{(3)}_{supp} = \{3, 1, 2\}$, $J_N = \{5, 6, 4\}$, the corresponding support matrix is

$$A(I, J^{(3)}_{supp}) = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix} \quad \text{since} \quad \det A(I, J^{(3)}_{supp}) \neq 0.$$

It easy to check that

$$A^{-1}(I, J^{(3)}_{supp}) = \begin{pmatrix} 1 & -2 & 1 \\ 0 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

The new feasible point $x^{(3)} = (2, 5, 4, 2, 0, 0)$. Calculate the potentials

$$u = c^T_{supp} A^{-1}_{supp} = (2, 3, 2) \cdot \begin{pmatrix} 1 & -2 & 1 \\ 0 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix} = (2, 0, 1)$$

then $\Delta = u^T A_N - c^T_N = (2, 0, 1) \cdot \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} - (0, 0, 0) = (0, 1, 2) \Longrightarrow \Delta^T = (0, 0, 0, 2, 0, 1)$. Clear, that optimality conditions are not satisfied:

$$\begin{aligned} \Delta_6 = 1 > 0 \quad & \text{for} \quad (x^{(3)}_6 = 0) = (d_{*6} = 0), \checkmark \\ \Delta_5 = 0 \quad & \text{for} \quad (x^{(3)}_5 = 0) = (d_{*5} = 0), \checkmark \\ \Delta_4 = 2 > 0 \quad & \text{for} \quad (x^{(3)}_4 = 2) \neq (d_{*4} = 0), \maltese \;\; \{6, 5, 4\} \in J_N \end{aligned}$$

2) Find the new feasible point $\bar{x}^{(3)} = x^{(3)} + \theta^o \cdot l_a$, where

$$l_a = z - x, \text{ and } \quad z = \begin{cases} d_{*j}, & if \ \ \Delta_j > 0; \\ d^*_j, & if \ \ \Delta_j < 0, \ j \in J_N; \\ d_{*j} \text{ or } \ d^*_j, & if \ \ \Delta_j = 0. \end{cases} \quad \Longrightarrow$$

$$\begin{cases} z_6 = 0, & \text{since} \ \ \Delta_6 = 1 > 0 \\ z_5 = 0, & \text{since} \ \ \Delta_5 = 0 \\ z_4 = 0, & \text{since} \ \ \Delta_4 = 2 > 0 \end{cases}$$

$$\Downarrow$$

$$z_{supp} = A^{-1}_{supp}(b - A_N z_N) = \begin{pmatrix} 1 & -2 & 1 \\ 0 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix} \cdot \left[\begin{pmatrix} 8 \\ 7 \\ 12 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right] = \begin{pmatrix} 6 \\ 2 \\ 5 \end{pmatrix}$$

Then the supporting components of the pseudo-feasible point $z = (z_{supp}, z_N)$ is

$$z_{supp} = (z_3 = 6, \quad z_1 = 2, \quad z_2 = 5.)$$

It is easy to check that its satisfies to the prime constraints $d^*_{supp} \leq z_{supp} \leq d{*}_{supp}$ by supporting components, namely :

$$\begin{aligned}
d_{*3} = 0 \leq \quad z_3 = 6 \quad &\leq d^*_3 = \infty(8) \qquad \checkmark \\
d_{*1} = 0 \leq \quad z_1 = 2 \quad &\leq d^*_1 = \infty(8) \qquad \checkmark \\
d_{*2} = 0 \leq \quad z_2 = 5 \quad &\leq d^*_5 = \infty(8) \qquad \checkmark
\end{aligned}$$

The conditions above satisfies automatically for non-supporting components of of pseudo feasible point $z = (z_{supp}, z_N)$ by construction. Thus the constructed vector z is optimal one!

Thus we obtain the optimal solution

$$z = \begin{pmatrix} 2 \\ 5 \\ 6 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \Longrightarrow \qquad x^{\boldsymbol{optimal}} = \begin{pmatrix} 2 \\ 5 \\ 6 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

It is easy to check that constructed vector $x^{\boldsymbol{optimal}}$ satisfies to the conditions of the theorem 4:

$$\begin{cases}
\Delta_j \geq 0 \ \text{ for } x_j = d_{*j}, \\
\Delta_j \leq 0 \ \text{ for } x_j = d^*_j, \\
\Delta_j = 0 \ \text{ for } \ d_{*j} \leq x_j \leq d^*_j, \quad j \in J_N
\end{cases}$$

The vector of estimates $\Delta^T = (0, 0, 0, 2, 0, 1)$ then:

$$\begin{aligned}
\Delta_1 = 0 \quad &\Longrightarrow \quad x_1 = 2in[0, \infty] \qquad \checkmark \\
\Delta_2 = 0 \quad &\Longrightarrow \quad x_2 = 5 \in [0, \infty] \qquad \checkmark \\
\Delta_3 = 0 \quad &\Longrightarrow \quad x_3 = 6 \in [0, \infty) \qquad \checkmark \\
\Delta_4 = 2 > 0 \quad &\Longrightarrow \quad x_4 = 0 = d_{*4} \qquad \checkmark \\
\Delta_5 = 0 \quad &\Longrightarrow \quad x_5 = 0 \in [0, \infty) \qquad \checkmark \\
\Delta_6 = 1 > 0 \quad &\Longrightarrow \quad x_6 = 0 \in [0, \infty) \qquad \checkmark
\end{aligned}$$

Thus we have been realized the principle of the decreasing the suboptimality estimates on iteration of the method , namely, after the first iteration

$\beta(x^{(1)}, J^{(1)}_{supp}) = 26$, (the exact value of deviation from optimum was $28-14 = 14$).

Then

$$\beta(\bar{x}^{(1)}, \bar{J}^{(1)}_{supp}) = \beta(x^{(2)}, J^{(2)}_{supp}) = \frac{21}{2} = 10.5, \qquad (28 - 20.5 = 7.5)$$

Then $\beta(x^3, J^{(3)}_{supp}) = \frac{11}{2} = 5.5$ (the exact value of deviation from optimum was $28-24 = 4$). Finally we obtain

$$\beta(x^{optimal}, J^{(3)}_{supp}) = 0$$

5.5 Summary of the adaptive method

A few observations about typical simplex and adaptive implementations can now be made.

- The basic "instrument" of the adaptive method - support - quite flexible react on a different situation during the solution process.
- Simplex methods start from a specified basis (B and x). The support lets us satisfy the general constraints $Ax = b$ initially and later.
- Nonsupport (nonbasic) variables need not be zero—they may have any value satisfying the bounds.
- The adaptive method allows to use any priory information about feasible solution.
- The new principle used on iteration of the adaptive method.
- The method equipped with stop criteria.
- The primal adaptive method significantly uses the ideas of the dual theory.(dual step in second procedure)
- The dual adaptive method is much more effective then traditional dual simplex methods due to the long step rule.
- ....provide sensitivity analisys

  Numerical experiments was realized in Matlab. The main criterion for comparison of the primal methods there was CPU solution time and number of iteration for the different values of m and n.

$m \times n$	30×45		70×100	
Problem number :	1	2	1	2
The number of iterations (adaptive)	18	13	48	54
The number of iterations (simplex)	21	18	58	58

The dual methods has been tested in [Kostina 2000]

An adaptive methods as well as simplex methods are iterative, finite, exact(satisfied all constraints) and relaxed(in a sense of the value of objective function). Thus in some sense the adaptive method is analog of simplex method, but the ideas of adaptive method is more naturally can be applied to the dynamical LP problems (also known as a discrete optimal control problems).

Chapter 6

Dynamical programming approach

It is known that dynamical programming method can be used for optimization problems. The method is attributed to Professor R.Bellman (USA) (see Bellman, 1957). Its continuous-time version is very broad generalization of classical Hamilton-Jacobi techniques to variational problems of control. The method is connected with embedding the construction of the optimal process into a family of identical problems with arbitrary initial conditions. This requires that the control would depend both on time and the state space variable, being presented in feedback (closed loop) form. The first indications on engineering solutions to specific problem of control synthesis were given by Flugge-Lotz in Germany, D.W. Bushaw in USA and A.A. Feldbaum in USSR. In fact, the work of Feldbaum on feedback control for automation served as an applied motivation for the development of Pontryagin Maximum Principle. In fact, a considerable amount of research was fullfilled by research groups at the Institute of Control Problem in Moscow. Applied problems of flight control were investigated by A.M. Letov, B.T. Polyak, Y.Z. Tsypkin.

6.1 Minimal path track for single UAV

Assume that we are need to provide the observation of $Z_2, ..., Z_n$ zones by single UAV Z_1. These object can be represented by the collection of n points from $\mathbb{R}^2$. It is supposed initially that the endurance T of the given UAV is sufficient to visit the given set of zones. The problem is to construct a shortest pass way from Z_1 to all zones $Z_2, ..., Z_n$ and come back to Z_1.

Solution by dynamical programming method

According to this method first we are needed to realize the so-called invariant embedding of the problem into parametric set of the similar problems. In our case instead of the fixed number n of zones we will consider the group composed by arbitrary s zones $1 \leq s \leq n-1$. That is the each group of s zones will be constitutes from arbitrary $Z_{i_1}, Z_{i_2}, ..., Z_{i_s}$ zones from the given collection. Moreover, instead of the pre-assigned initial point Z_1 consider the arbitrary zone Z_i from the given collection $Z_1, Z_2, Z_3, ..., Z_n$.

Next define the so-called Bellman function $B_s(i|i_1, i_2, ..., i_s)$ as a shortest pass way between the point Z_i and Z_1, and trough-passing the given zones $Z_{i_1}, Z_{i_2}, ..., Z_{i_s}$.

In order to construct the equation such that the Bellman function will satisfy this equation, we will make the trial moving from point Z_i to point Z_{i_k} where $k = 1, ..., s$. Then, in accordance with the definition of Bellman function given above, the shortest pass way from Z_{i_k} to Z_1 trough the zones $Z_{i_1}, ..., Z_{i_{k-1}}, Z_{i_{k+1}}, ..., Z_{i_s}$ (note that the zone Z_{i_k} is absent in this collection) is denotes as

$$B_{s-1}(i_k|i_1, ..., i_{k-1}, i_{k+1}, ..., i_s).$$

Hence the length of a route from Z_i to Z_1 will be not less then

$$d_{ii_k} + B_{s-1}(i_k|i_1, ..., i_{k-1}, i_{k+1}, ..., i_s)$$

where d_{ij} denotes here the distance between Z_i and Z_j.

Therefore, the minimal route from $Z_i \longrightarrow$ to Z_1 is

$$B_s(i|i_1, i_2, ..., i_s) = \min_{1 \leq k \leq s} \left[d_{ii_k} + B_{s-1}(i_k|i_1, ..., i_{k-1}, i_{k+1}, ..., i_s) \right] \tag{6.1}$$

and it is defined from the the expression $d_{ii_k} + B_{s-1}(i_k|i_1, ..., i_{k-1}, i_{k+1}, ..., i_s)$ by enumeration of possibilities of all trial steps from Z_i to Z_{i_k}, where $k,\ 1 \leq k \leq s$. The boundary condition for the Bellman equation (6.1) follows immediately from the definition of the Bellman function at $s = 1$:

$$B_1(i|i_1) = d_{ii_1} + d_{i_1 1} \tag{6.2}$$

Using the initial conditions (6.2) and the equation (6.1) we can calculate sequentially the functions

$$B_1(i|i_1), B_2(i|i_1, i_2), ..., B_{n-2}(i|i_1, i_2, ..., i_{n-2}), B_{n-1}(i|i_1, i_2, ..., i_{n-1}).$$

In final, the obtained the function value

$$B_{n-1}(1|2, ..., n) \tag{6.3}$$

presents the length of the shortest pass way for the considered optimization problem.

Next, the optimal pass way can be designed by the following procedure. From (6.1) we find for $s = n - 1$

$$\begin{aligned} B_{n-1}(1|2,3,..,n) = \min_{2\le i\le n}[d_{1i} + B_{n-2}(i|2,3,...,i-1,i+1,...,n)] = \\ = d^0_{1i^0} + B_{n-2}(i^0|2,3,...,i^0-1,i^0+1,...,n) \end{aligned} \tag{6.4}$$

Note that the last equality denotes that the minimum in (6.4) is reached on the index i^0. This gives that the first zone of optimal pass way from Z_1 is the zone Z_{i^0}.

Further, again from (6.1) we find at $s = n - 2$:

$$\begin{aligned} B_{n-2}(i^0|2,3,...,i^0-1,i^0+1,...,n) = \\ \min_{i\in[2,...,n]\setminus\{i^0\}}\left[d_{i^0i} + B_{n-3}(i^0|2,3,...,i-1,i+1,...,i^0-1,i^0+1,...,n)\right] = \\ = d_{i^0i^1} + B_{n-3}(i^0|2,3,...,i^1-1,i^1+1,...,i^0-1,i^0+1,...,n) \end{aligned} \tag{6.5}$$

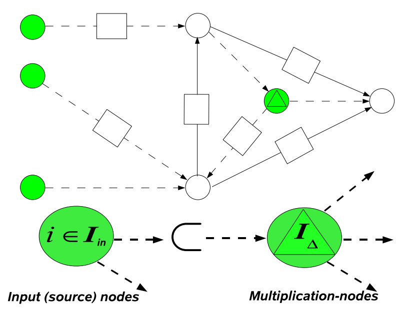
Thus the optimal pass way is continued by zones Z_{i^1} such that the optimal route is

$$Z_1 \longrightarrow Z_{i^o} \longrightarrow Z_{i^1} \longrightarrow (etc).$$

The described procedure is repeated next by the obvious manner.

Example

The problem is to find the shortest pass way for UAV of Z_1 to service the set of zones Z_2, Z_3, Z_4, the mutual distances between of which are given by the following Table A and imaged by the graph



	z_1	z_2	z_3	z_4
z_1	0	1	2	3
z_2	1	0	3	1
z_3	2	3	0	4
z_4	3	1	4	0

Table A

The solution of the given problem find by the dynamic programming method described above. The definition of Bellman function and (6.3) show that the minimal value of the desired pass way is equal to $B_3(1|2,3,4)$. By definition of the function $B_3(1|2,3,4)$ we have:

$$B_3(1|2,3,4) = \min\Big(d_{12} + B_2(2|3,4); d_{13} + B_2(3|2,4); d_{14} + B_2(4|2,3)\Big) \tag{6.6}$$

Hence we have to know the values $B_2(2|3,4), B_2(3|2,4), B_2(4|2,3)$ that are defined as follows:

$$B_2(2|3,4) = \min\left(d_{23} + B_1(3|4); d_{24} + B_1(4|3)\right) \tag{6.7}$$

$$B_2(3|2,4) = \min\left(d_{32} + B_1(2|4); d_{34} + B_1(4|2)\right) \tag{6.8}$$

$$B_2(4|2,3) = \min\left(d_{42} + B_1(2|3); d_{43} + B_1(3|2)\right) \tag{6.9}$$

Again we have to know the values $B_1(3|4); B_1(4|3); B_1(2|4); B_1(4|2); B_1(2|3); B_1(3|2)$. Find them in accordance with (6.2) and the distances given by the table (graph):

$$B_1(3|4) = 7; B_1(4|3) = 6;$$

$$B_1(2|4) = 4; B_1(4|2) = 2;$$

$$B_1(2|3) = 5; B_1(3|2) = 4$$

Now we are able to find the needed values of Bellman function:

$$B_2(2|3,4) = \min\left(3+7; 1+6\right) = 7$$

$$B_2(3|2,4) = \min\left(3+4; 4+2\right) = 6$$

$$B_2(4|2,3) = \min\left(2+5; 4+4\right) = 7$$

and finally

$$B_3(1|2,3,4) = \min\left(1+7, 2+6, 3+7\right) = \min\left(8,8,10\right) = 8$$

Thus, the minimal length of the pass way is equal 8.

Note, that the optimal value is reached on two variants

$$8 = B_3(1|2,3,4) = \begin{cases} d_{12} + B_2(2|3,4) \\ d_{13} + B_2(3|2,4) \end{cases} \tag{6.10}$$

For brevity sake, choose the second variant. That is the minimum in (6.6) (see, also, (6.4)) is reached on the index $i^0 = 3$ such that

$$B_3(1|2,3,4) = d_{13} + B_2(3|2,4) = 8 \tag{6.11}$$

This means that the first zone to be visit is $Z_1 \to Z_3$.

From (6.11) follows that further procedure is to consider the value of $B_2(3|2,4)$. As follows from (6.8)

$$B_2(3|2,4) = \min\left(d_{32} + B_1(2|4); d_{34} + B_1(4|2)\right) = \min\left(3+4; 4+2\right) = 6. \tag{6.12}$$

Since the minimum in the last is reached on the value $d_{34} + B_1(4|2)$ then the next index for the desired optimal pass way is $i^1 = 4$ (see, also (6.5)). Hence the next zone to visit is Z_4 such that the optimal pass is continued as $Z_1 \to Z_3 \to Z_4$. It is obviously that the last remained zone to be visited is Z_2. Hence the optimal route is

$$Z_1 \to Z_3 \to Z_4 \to Z_2 \to Z_1.$$

Obviously, the back pass

$$Z_1 \to Z_2 \to Z_4 \to Z_3 \to Z_1$$

is optimal, too. Note that this fact is detected by (6.10), if the first variant for minimization will be chosen in order to continue the given procedure.

The considered problem is simplest one and can be solved by usual examination of options. Since for the set of indexes $\{2,3,4\}$ (these indexes correspond to the numeration of zones Z_2, Z_3, Z_4)) there is $3! = 1 \cdot 2 \cdot 3 = 6$ variants. They and their total route length are as follows

$$\begin{aligned}
1 \to 2 \to 3 \to 4 \to 1 &= 1+3+4+3 = 11\\
1 \to 3 \to 2 \to 4 \to 1 &= 2+3+1+3 = 9\\
1 \to 4 \to 3 \to 2 \to 1 &= 3+4+3+1 = 11\\
1 \to 2 \to 4 \to 3 \to 1 &= 1+1+4+2 = 8\\
1 \to 3 \to 4 \to 2 \to 1 &= 2+4+1+1 = 8\\
1 \to 4 \to 2 \to 3 \to 1 &= 3+1+3+2 = 9.
\end{aligned} \tag{6.13}$$

Again, we have the optimal track as $Z_1 \to Z_3 \to Z_4 \to Z_2 \to Z_1$ or $Z_1 \to Z_2 \to Z_4 \to Z_3 \to Z_1$.

6.1.1 Minimal path track for the single UAV with restricted endurance

The method described above can be used for the case when the operational resource of UAV (endurance) is not sufficient to serve all the pre-assigned zones. Noting that by the definition of Bellman function the value $B_s(i|i_1, i_2, ..., i_s)$ is a shortest pass way between the point Z_i and Z_1, and trough-passing the given zones $Z_{i_1}, Z_{i_2}, ..., Z_{i_s}$ yields that the maximal number t^* of zones available for the visits from Z_1 and come back to Z_1 is the greatest number satisfying the following inequality

$$t^* : \quad B^* \doteq \max_t B_t(i|i_1, i_2, ..., i_t) \leq L \tag{6.14}$$

where L is maximal path length available for the given UAV.

Then the optimal collection zones to be visited is determined on the base of analysis of the term

$$B_{t^*}(1|i_1, i_2, ..., i_{t^*}) = B^*, \tag{6.15}$$

on which the last maximum is achieved. This analysis is realized by analogy with the Section above.

6.2 Optimal endurance allocation of UAVs for multiple zones

Consider the case when we have a set of UAV the total endurance of which is restricted by some value T. Let $Z_1, ..., Z_n$ be the zones that are required to serve. The endurance T (the total flying time of the given group of UAVs, for example) can be treated as a "resource" of the given UAVs. Then the problem is to distribute the given resource Tamong n zones.

Let $t_i, 0 \leq t_i \leq T$ is the portion of "resource" assigned for the zone Z_i. Then the "benefit" of this assignment we denote by $f_i(t_i)$. The benefit here can be treated as the probability of targets detection in i-th zone or the square D_i of observed area in i-th zone, for example.
1) We are assume that we have already some calculations of probability of targets detection depending on time.
2) Also it is possible to calculate the square of observed area in the zone in the given time.
3) For simplicity also assume that the time necessary to fly from zone to zone is not essential.

The problem statement are:

$$\sum_{i=1}^{n} f_i(t_i) \to \max \qquad \text{subject to} \quad \sum_{i=1}^{n} t_i \le T,\ t_i \ge 0, i = 1, ..., n \tag{6.16}$$

Note that the specific feature of the problem (6.16) is separable form of the cost value function

In this paper the dynamic programming method is adopted for solution of the considered optimization problem.

First, this problem should be embed into the collection $P(k, y)$ of parametric problems of the form

$$P(k,y): \qquad \sum_{i=2}^{k} f_i(t_i) \to \max, \qquad \text{subject to} \quad \sum_{i=1}^{k} t_i \le y,\ t_i \ge 0, i = 1, ..., k \tag{6.17}$$

where $0 \le y \le T,\ \ 1 \le k \le n$ are the parameters of the given collection. Obviously, that the initial problem can be obtained from this collection at $y = T$ and $k = n$.

Introduce now the Bellman function $B_k(y)$ as follows

$$B_k(y) = \max_{t_i} \sum_{i=2}^{k} f_i(t_i), \qquad \text{subject to} \quad \sum_{i=2}^{k} t_i \le y,\ t_i \ge 0, i = 1, ..., k \tag{6.18}$$

Let $z,\ 0 \le z \le y$ be the portion of the endurance assigned for the zone Z_k. The corresponding cost value (”benefit”) of such distribution is equal $f_k(z)$. Hence, the rest endurance $y - z$ should be distributed among the remained $(k-1)$ zones $Z_1, \ldots, Z_{k-1}$. In accordance with the definition of Bellman function the optimal distribution of the endurance $y - z$ among $(k-1)$ zones is determined as $B_{k-1}(y-z)$. Therefore, if the given resource is equal y, then after the assignment of z portion for zone Z_k the total profit of all k zones is

$$f_k(z) + B_{k-1}(y-z) \tag{6.19}$$

Hence, the optimal distribution z_k^0, $0 \le z_k^0 \le y$, for the given zone Z_k is determined by the following condition

$$f_k(z_k^0) + B_{k-1}(y - z_k^0) = \max_{0 \le z \le y} \Big[f_k(z) + B_{k-1}(y-z) \Big] \tag{6.20}$$

But in accordance with the definition (6.18) maximal profit from distribution of the initially given resource y among all k zones is equal $B_k(y)$. Thus, for the introduced Bellman function $B_k(y)$ we have the following Bellman equation

$$B_k(y) = \max_{0 \le z \le y} \Big[f_k(z) + B_{k-1}(y-z) \Big] \tag{6.21}$$

The initial condition for the obtained recurrent Bellman equation (6.21) follows from (6.18) at $k = 1$:

$$B_1(y) = \max f_1(t_1), \qquad \text{subject to} \quad t_1 = y,\ t_1 \geq 0$$

Hence the initial condition for the Bellman equation (6.21) is

$$B_1(y) = f_1(y). \tag{6.22}$$

Now we can design an optimal solution for the allocation problem. Put $k = 2$ in the equation (6.21):

$$B_2(y) = \max_{0 \leq z \leq y} \Big[f_2(z) + B_1(y - z) \Big] = \max_{0 \leq z \leq y} \Big[f_2(z) + f_1(y - z) \Big] \tag{6.23}$$

Since f_1, f_2 are known functions then using the initial conditions (6.22) we are able calculate the maximum in (6.24) and, hence, find the function $B_2(y)$. Put $k = 3$ in the equation (6.21):

$$B_3(y) = \max_{0 \leq z \leq y} \Big[f_3(z) + B_2(y - z) \Big] = \max_{0 \leq z \leq y} \Big[f_3(z) + f_2(y - z) + f_1(2y - z) \Big] \tag{6.24}$$

Since f_3 and B_2 are known functions, we can calculate $B_3(y)$. Proceeding sequentially this procedure we will determine the functions

$$B_4(y), B_5(y), ..., B_n(y).$$

In final, the function value

$$B_n(T) \tag{6.25}$$

presents the maximal profit for the initial endurance allocation problem (6.16).

In order to design the optimal distribution of the available endurance of UAV we realize the reverse motion in the procedure of solution of Bellman equation.

Put in (6.20) $k = n,\ y = T$ and find the value $t_n^0 \doteq z^0(T)$ on which the maximum in (6.20) is achieved for the given data. According to the definition (6.18)of Bellman function this value means the optimal portion of the endurance allocated for the zone Z_n.

Further, if for the zone Z_n the portion t_n^0 of the initial endurance T was allocated then the rest $T - t_n^0$ of endurance are available to allocate among the remainder zones $Z_1, \ldots, Z_{n-1}$. Put in (6.20) $k = n - 1,\ y = T - t_n^0$ and find the value $t_{n-1}^0 \doteq z^0(T - t_n^0)$ on which the maximum in (6.20) is achieved for the given data. Again in accordance with the definition (6.18)of Bellman function the obtained value means that the optimal portion of the remainder endurance allocated for the zone Z_{n-1} is equal t_{n-1}^0. Ongoing analogously this procedure gives the desired optimal allocation of the endurance T among the given zones $Z_n, Z_{n-1}, \ldots, Z_1$ in the form

$$t_n^0, t_{n-1}^0, \ldots, t_2^0, t_1^0.$$

Illustrative example

Let Z_1, Z_2, Z_3 be the zones under service of five UAVs. Assume that the "benefits" of their service in each zone are given in Table 1. For example, the "benefit" of service in the given zone for each UAV can be interpreted as the number of the detected targets. Then $f_i(x)$ means the "benefit" of using x UAVs for service of zone Z_i,where x denotes the number of the used UAV.

x	0	1	2	3	4	5
$f_1(x)$	0	1	2	3	4	5
$f_2(x)$	0	0	1	2	4	7
$f_3(x)$	0	2	2	3	3	5

Table 1

The problem is to find an optimal allocation of the given UAVs to serve the given zone Z_1, Z_2, Z_3 such that to maximize the total "benefit".

For the considered example we have $n = 3$ and $T = 5$. According to the definition of Bellman function (6.18) the optimal solution of the considered allocation problem is determined by the function value

$$\begin{aligned}
&B_3(5) = \max_{0\leq z\leq 5} \Big[f_3(z) + B_2(5-z)\Big] = \\
&\max\Big\{f_3(0) + B_2(5); f_3(1) + B_2(4); f_3(2) + B_2(3); \\
&f_3(3) + B_2(2); f_3(4) + B_2(1); f_3(5) + B_2(0)\Big\}
\end{aligned} \tag{6.26}$$

where the required here the functions $B_2(y)$ are determined by the following formula

$$B_2(y) = \max_{0\leq z\leq y} \Big[f_2(z) + B_1(y-z)\Big] \tag{6.27}$$

Since the initial conditions gives $B_1(y) = f_1(y)$ then from Table 1 follows

$$B_1(0) = 0,\ B_1(1) = 1,\ B_1(2) = 2,\ B_1(3) = 3,\ B_1(4) = 4,\ B_1(5) = 5.$$

Using the equation (6.27) and the obtained $B_1(y)$ we can find the required function values

$$B_2(1), B_2(2), ..., B_2(5).$$

For example,

$$
\begin{aligned}
B_2(4) &= \max_{0\le z\le 4}\Big[f_2(z)+B_1(4-z)\Big] = \\
&= \max\Big\{f_2(0)+B_1(4); f_2(1)+B_1(3); f_2(2)+B_1(2); \\
& f_2(3)+B_1(1); f_2(4)+B_1(0)\Big\} =
\end{aligned}
\tag{6.28}
$$

$$
= \max\Big\{0+4; 0+3; 1+2; 2+1; 4+0\Big\} = 4
$$

It is convenient the obtained values of the functions $B_2(y)$ ($B_1(y)$ and $B_3(y)$, also) to collect in the Table 2. Also, in order to simplify the next calculation the obtained Bellman function values $B_k(y)$ is accompanied (in braces) by the arguments $z_k^0(y)$ at which this value is achieved (that is the right hand side of (6.27) reaches their maximum). Using the obtained values $B_2(1), B_2(2), ..., B_2(5)$ we are able to end (6.26) as follows

$$
\begin{aligned}
& B_3(5) = \\
& \max\Big\{f_3(0)+B_2(5); f_3(1)+B_2(4); f_3(2)+B_2(3); \\
& f_3(3)+B_2(2); f_3(4)+B_2(1); f_3(5)+B_2(0)\Big\} =
\end{aligned}
\tag{6.29}
$$

$$
= \max\Big\{0+7; 2+4; 2+3; 3+2; 3+1; 5+0\Big\} = 7.
$$

Note, that by analogy with this calculations we can find the other function values $B_3(y)$ that are presented in Table 2.

y	0	1	2	3	4	5
$B_1(y)$	0	1	2	3	4	5
$B_2(y)$	0	1(0)	2(0)	3(0)	4(0,4)	7(5)
$B_3(y)$	0	2(1)	3(1)	4(1)	5(1)	7(0)

Table 2

Thus, the the maximal profit available in the considered example is equal $B_3(5) = 7$. The corresponding optimal allocation is designed on the base of analysis of the obtained maximal Bellman function value $B_3(5) = 7$;

$$
7 = B_3(5) = f_3(0) + B_2(5), \tag{6.30}
$$

The last denotes that the maximal efficiency is achieved at $t_3^0 \doteq z^0(5) = 0$. This means that no UAVs are not directed for service of the zone Z_3. Hence, the rest $5-0=5$ of UAVs should be optimal allocated among the remainder zones Z_2 and Z_1.

Again, accordingly to the Definition of Bellman function (6.18) the maximal efficiency in this case is equal $B_2(5)$, and Table 2 gives that this value is $B_2(5) = 7$ that is achieved at $t_2^0 \doteq z^0(5) = 5$. This means that five UAVs should be directed to the zone Z_2 to get the maximal profit. Hence, for the zone Z_1 is no UAVs for their service.

In final, the optimal distribution of five UAVs for the considered allocation problem with the given initial data is

$$t_1^0 = 0,\ t_2^0 = 5,\ t_3^0 = 0.$$

The obtained solution can be easy find by the careful analysis of Table 1, where the maximal efficiency value 7 for the case of the total endurance $T = 5$ is given by the allocation of all 5 UAVs to zone Z_2.

But some changing initial data leads to a nontrivial optimization problems. For example, let the initial group formed by 5 UAVs is decreased to four UAVs such that $T = 4$. In this case the maximal efficiency is equal 5. This fact follows from Table 2 where we have $B_3(4) = 5$. The corresponding optimal allocation of 4 UAVs among 3 zones is determined by the following: since the maximal value function $B_3(4) = 5$ is achieved at $t_3^0 \doteq z_3^0(4) = 1$ (this value is written in the braces of the corresponding cell) then one UAV is directed to zone Z_3. Further, the rest $4 - 1 = 3$ of UAVs should be allocated among the remainder zones Z_1, Z_2. The Table 2 gives the optimal value of efficiency $B_2(3) = 3$ which is achieved at $t_2^0 \doteq z_2^0(3) = 0$. This means that UAVs are not allocated to zone Z_2. Therefore, for service of the zone Z_1 we have $3 - 0 = 3$ UAVs.

Thus, for the case $T = 4$ the solution of the form

$$t_1^0 = 3,\ t_2^0 = 0,\ t_3^0 = 1$$

is the optimal allocation for the given 4 UAVs among 3 zones.

Note, that the search of optimal solution for $T = 4$ is a nontrivial in contrast the case $T = 5$. As it follows the remark mentioned above the being a "schedule" like to the Table 2 gives a good tool for the fast solution of the allocation problems with disturbed data.

6.3 Schedule representation for hierarchical missions of UAVs

In some cases, the hierarchical missions can be presented by the collection I of heterogeneous tasks that should be served by some group of UAVs. It is supposed that the given tasks should be served by UAVs in the pre-assigned order such that every next task cannot be served before the previous task is completed.

Let $S \triangleq \{I, U\}$ denotes a network, where I means the set of the nodes(tasks) and U is the set of arcs (possible services). The service of task $j \in I$ which is available only after the ending task $i \in I$ we will denote by the arc $(i,j) \in U$. Hence any service $(i,j) \in U$ is not available if the services $(k,i) \in U,\ k \in I_i^- \triangleq \{k \in I : \exists\ (k,i) \in U\}$ are not completed. The moment of the service end for the task i is determined by the moments of the **completion of all services** of $(k,i),\ k \in I_i^-$.

In the network $S = \{I, U\}$ choose two nodes s and t, where s denotes the starting node (that is the start of the mission) and t is the final node where the mission is completed.

Denote $I_i^+ = \{j \in I : \exists\ (i,j) \in U\}$. It is obvious that $I_s^- = \varnothing,\ I_t^+ = \varnothing$. Denote by c_{ij} the service time for the arc $(i,j) \in U$. Also, let $x_i, i \in I$ denotes the moment of the service completion for the task i.

The given request of the pre-assigned order in service leads to the following inequalities

$$x_i + c_{ij} \leq x_j,\ i \in I_j^-,\ j \in I. \tag{6.31}$$

These inequalities image the fact that the service of j cannot be realized before completion of all services of $(i,j),\ i \in I_j^-$ of the previous task i.

Then the minimal service time of the mission is determined as the smallest number x_t^0 that together the numbers $x_i^0 \geq 0,\ i \in I \setminus \{t\};\ x_s^0 = 0$ satisfy the inequalities (6.31).

Since for ending mission it is necessary to finish all of services then the length $\sum c_{ij}$ of each route $s \to t$ formed by the set of available arcs is not less x_t^0. Therefore, the search of the optimal solution x_t^0 is equivalently to the problem: find the route from the nodes s to the node t such that the length $\sum c_{ij}$ of this rout is **maximal**. The route with such kind of property is called **extremal route**.

The formalization of this problem can be given by the following

$$x_t - x_s \to \min \tag{6.32}$$

subject to

$$\begin{aligned} x_i + c_{ij} \leq x_j,\ i \in I_j^-,\ j \in I \\ x_i^0 \geq 0,\ i \in I \setminus \{t\} \end{aligned} \tag{6.33}$$

For solution of this optimization problem we use the dynamic programming method.

For this purpose embed this problem into the parametric collection of optimization problems when instead the concrete terminal node t we will consider an arbitrary node j. For the extended optimization problem introduce the Bellman function B_j as a largest pass way from the node s to the node j.

In order to get the corresponding Bellman equation to which the introduced function satisfies we investigate the pass way from s to j. Suppose that the last arc of this pass way is the arc (i, j) where $i \in I_j^-$ and, moreover, the previous pass way from s to i is optimal such that the length of this pass $s \to i$ is equal B_i. Then the total length of the pass way $s \to i \to j$ is equal $B_i + c_{ij}$. Therefore, the optimal (maximal) length of the pass $s \to j$ is given as $\max\limits_{i \in I_j^-}(B_i + c_{ij})$. Then from the definition of Bellman function follows immediately that the function B_j satisfies the following equation

$$B_j = \max_{i \in I_j^-}(B_i + c_{ij}) \tag{6.34}$$

The initial condition for this Bellman equation is given

$$B_s = 0 \tag{6.35}$$

To solve equation (6.34)–(6.35) we use the following approach. Denote by I_* the set of nodes from $i \in I$, for which the values of Bellman function B_i are known yet. The set I_* is not empty since $s \in I_*$, at least. If $t \in I_*$ then the considered optimization problem is solved since in this case B_t is optimal length of the pass $s \to t$, and the design of the corresponding pass way is realized by the analysis of the optimal value B_t by "backward motion" in Bellman equation. To simplify this "backward motion" for the nodes $i \in I_*$ introduce the additional function $f(i)$ $i \in I_*$. At the first stage we have

$$I_* = s, \ B_s = 0, \ f(s) = 0. \tag{6.36}$$

Let us now $t \notin I_*$. In the network $S = (I, U)$ we find the set of nodes $w(I_*)$ neighboring with the nodes of I_* :

$$w(I_*) = \{j \in I : (i, j) \in U, \ j \notin I_*, \ i \in I_*\}. \tag{6.37}$$

It can be shown that in the set $w(I_*)$ there exists the node $j_* \in w(I_*)$ such that $I_{j^*}^- \subset I_*$. Since for all nodes $i \in I_*$ the values of the Bellman function B_i are known then the Bellman equation (6.34) gives

$$B_{j^*} = \max_{i \in I_{j^*}^-}(B_i + c_{ij^*}) = c_{i^*j^*} + B_{i^*}. \tag{6.38}$$

Then the second stage of the procedure is to extend the set I_* and to calculate the associated function value $f(i)$ as

$$I_* \triangleq I_* \cup \{j_*\}, \ f(j_*) = i_*. \tag{6.39}$$

Note that the nodes of j_* satisfying the property $I_{j^*}^- \subset I_*$ can be nonunique. In this case the corresponding values of the functions $B_j, f(j)$ are determined by the same procedure of (6.38). Further the next iterations are continued by analogy with given above. It is obviously that the number of the described iterations does not exceed the number $|I|$ of elements in the set I.

In final, these iterations will lead to the case when $t \in I_*$. This means that the considered optimization problem is solved, and B_t is the optimal value of the length $s \to t$. The corresponding optimal " backward pass way"

$$\{t \leftarrow i_1 \leftarrow i_2 \leftarrow \ldots \leftarrow i_k \leftarrow s\}$$

is defined by the following formula

$$i_1 = f(t),\ i_2 = f(i_1), \ldots, i_k = f(i_{k-1}),\ s = f(i_k). \tag{6.40}$$

6.3.1 Illustrative example

Consider the complex mission given by the network of Figure 1 where $s = 1,\ t = 4$. Here the node 1 is the beginning of the mission. The nodes 2 and 6 image the tasks of the first submission, where there are two tasks in the node 2 and one task in the node 6. In addition, one of the tasks of the node 2 can be served only after the completion of the task in the node 6. Moreover, the service time of the first and the second task from the node 6 are equal 4 and 2, respectively, and the service time for the task of the node 6 is equal 1. The second submission is presented by two tasks in the node 3 and one task in the node 5, where the first task in the node 3 can be served after the completion of all tasks in the node 2, and the second task of 3 can be done after the completion of the task in 6. The corresponding service time are 2 and 4, respectively. The third submission is presented by two tasks in the node 5, where the first task can be done after the completion of the task in 6, and the second task is available after the ending tasks in 3. The corresponding service time are 1 for the both tasks. The node 4 presented by two tasks in the node 4 is the final submission. The service time of these tasks are 8 and 1.

The numbers marked under the arcs of Figure 1 denote the service time of the corresponding tasks of submissions.

Additionally, on the Figure 1 the Bellman function values $B_j,\ j = 1, \ldots, 6$ and function $f(j),\ j = 1, \ldots, 6$ calculated in accordance with (6.36)–(6.40) are marked. These calculations are realized by the following step-by-step procedure.

1) For $s = 1$ we have $I_*^{(1)} = \{s\} = \{1\}$ and $B_s = B_1 = 0,\ f(s) = f(1) = 0$. Then the

set

$$w(I_*^{(1)}) = \{j \in I : (i,j) \in U,\ j \notin I_*,\ i \in I_*\}$$

in this case is

$$w(I_*^{(1)}) = \{j \in I : (1,j) \in U\} = \{(1,2) \in U, (1,6) \in U\} = \{2;6\}. \tag{6.41}$$

Find now the node $j_* \in w(I_*^{(1)})$ such that $I_{j^*}^- \subset I_*^{(1)}$ where the set I_i^- is defined as $I_i^- \triangleq \{k \in I : \exists \text{ arc } (k,i) \in U\}$. In the considered case the needed node is unique $j_* = \{6\}$ because the corresponding set $I_6^- = \{k \in I : \exists \text{ arc } (k,6) \in U\} = \{k = 1 : \exists \text{ arc } (1,6) \in U\} = \{1\}$ satisfies the condition $\{1\} = I_6^- \subset I_*^{(1)} = \{1\}$. Note, that the set

$$I_2^- = \{k \in I : \exists \text{ arc } (k,2) \in U\} =$$

$$= \{k = 1, k = 6 : \exists \text{ arc } (1,2) \in U, (6,2) \in U\} = \{1;6\} \not\subset I_*^{(1)} = \{1\},$$

and hence, the node $j_* = \{2\}$ is not suitable for the choice.

Thus, the associated Bellman function value of

$$B_{j^*} = \max_{i \in I_6^-}(B_i + c_{ij^*}) = c_{i^*j^*} + B_{i^*}.$$

in the considered case is equal

$$B_6 = \max_{i \in I_6^-}(B_i + c_{i6}) = c_{16} + B_1 = 1 + 0 = 1 \tag{6.42}$$

where $i_* = \{1\}$ and the corresponding function $f(j_*) = i_*$ is equal $f(6) = 1$.

2) Extend the set $I_*^{(2)} \triangleq I_*^{(1)} \cup \{j_*\} = \{1;6\}$ and find

$$w(I_*^{(2)}) = w(\{1;6\}) = \{j \in I : (i,j) \in U,\ j \notin I_*^{(2)},\ i \in I_*^{(2)}\} =$$

$$= \{(1,2) \in U, (6,2) \in U, (6,3) \in U, (6,3) \in U\} = \{2;3;5\}.$$

It can be checked that

$$I_2^- = \{k \in I : \exists \text{ arc } (k,2) \in U\} =$$

$$= \{k = 1, k = 6 : \exists \text{ arc } (1,2) \in U, (6,2) \in U\} = \{1;6\} \subset I_*^{(2)} = \{1;6\}$$

such that the required node from the set $w(I_*^{(2)})$ is $j_* = \{2\}$. Note that the nodes $j_* = \{3\}$ and $j_* = \{5\}$ from the considered set $w(I_*^{(2)})$ are not suitable for the choice since the corresponding sets

$$I_3^- = \{k \in I : \exists \text{ arc } (k,3) \in U\} =$$

$$= \{k = 2, k = 6 : \exists \text{ arc } (2,3) \in U, (6,3) \in U\} = \{2;6\} \not\subset I_*^{(2)} = \{1;6\}$$

and

$$I_5^- = \{k \in I : \exists \text{ arc } (k,5) \in U\} =$$

$$= \{k = 3, k = 6 : \exists \text{ arc } (3,5) \in U, (6,5) \in U\} = \{3;6\} \not\subset I_*^{(2)} = \{1;6\}$$

do not satisfy the condition $I_{j*}^- \subset I_*^{(2)}$.

Hence

$$B_2 = \max_{i \in I_2^-}(B_i + c_{i2}) = \max\{c_{12} + B_1; c_{62} + B_6\} = \max\{2+0; 4+1\} = 5 \qquad (6.43)$$

Moreover, the corresponding elements are $i_* = \{6\},\ f(j_*) = f(2) = i_* = 6$.

3) Extend the set $I_*^{(3)} \triangleq I_*^{(2)} \cup \{j_*\} = \{1;6;2\}$ and find

$$w(I_*^{(3)}) = w(\{1;6;2\}) = \{j \in I : (i,j) \in U,\ j \notin I_*^{(3)},\ i \in I_*^{(3)}\} =$$

$$= \{(2,3) \in U, (6,3) \in U, (6,5) \in U\} = \{3;5\}.$$

It can be checked that

$$I_3^- = \{k \in I : \exists \text{ arc } (k,3) \in U\} =$$

$$= \{k = 2, k = 6 : \exists \text{ arc } (2,3) \in U, (6,3) \in U\} = \{2;6\} \subset I_*^{(3)} = \{1;6;2\}$$

such that the required node from the set $w(I_*^{(3)})$ is $j_* = \{3\}$. Note that the node $j_* = \{5\}$ from the considered set $w(I_*^{(3)})$ is not suitable for the choice since the corresponding set

$$I_5^- = \{k \in I : \exists \text{ arc } (k,5) \in U\} =$$

$$= \{k = 3, k = 6 : \exists \text{ arc } (3,5) \in U, (6,5) \in U\} = \{3;6\} \not\subset I_*^{(3)} = \{1;6;2\}$$

does not satisfy the condition $I_{j*}^- \subset I_*^{(3)}$. Hence

$$B_3 = \max_{i \in I_3^- = \{2;6\}}(B_i + c_{i3}) = \max\{c_{23} + B_2; c_{63} + B_6\} = \max\{2+5; 4+1\} = 7. \quad (6.44)$$

Moreover, the corresponding elements are $i_* = \{2\},\ f(j_*) = f(3) = i_* = 2$.

The remainder elements for nodes 5 and 4 are calculated by analogy with given above, and it is omitted here.

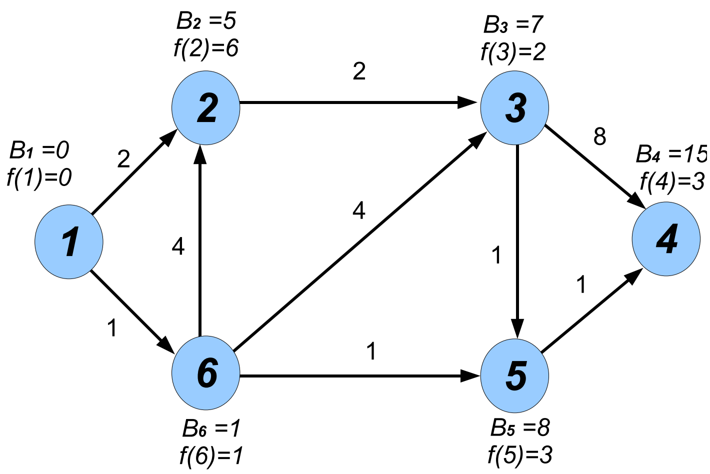


Figure 1

From this Figure it follows that the minimal time to serve mission is equal $B_4 = 15$. The obtained values of the function $f(j)$ make it possible to re-construct the optimal pass way $1 \to 4$ by the formula (6.40):

$$i_1 = f(t) = 3,\ i_2 = f(i_1) = f(3) = 2,\ i_3 = f(i_2) = f(2) = 6,\ i_4 = f(i_3) = f(6) = 1 \quad (6.45)$$

Hence, the optimal order to serve the tasks of the given mission is

$$s = 1 \to 6 \to 2 \to 3 \to 4 = t.$$

Note that the tasks of each node from the extremal route $(i, j,) \in U_{extrem}$ should be started exactly at the calculated moments $x_i^0, i \in I_{extrem}$. But the tasks of the nodes which are out the extremal route can be served early. Let $(i, j) \notin U_{extrem}$. Then the tasks in nonextremal node j can be served starting at any moment of the form

$$x_i^0 + \Delta t_{ij},\ \text{where}\ \Delta t_{ij} \in [0, x_j^0 - c_{ij} - x_i^0]$$

such that the optimal service time of whole mission is not changed. It should be emphasized that the given ability "to shift" the start for service of some tasks can be used to minimize the number of the UAVs involved in missions. In particular, since the moments $x_i^0, i \in I$ is determined then the optimization problem can be formulated as follows: find the moments $\Delta t_{ij},\ (i, j) \in U$ to minimize the number of UAVs needed to complete the all tasks of the given mission with minimal service time.

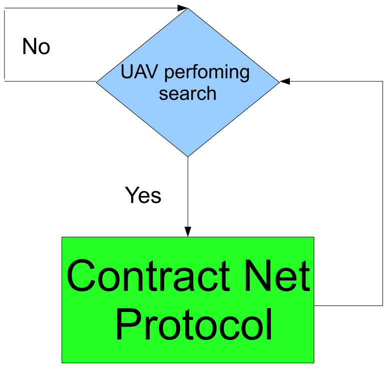
Chapter 7

Contract Net Protocol

7.1 Introduction

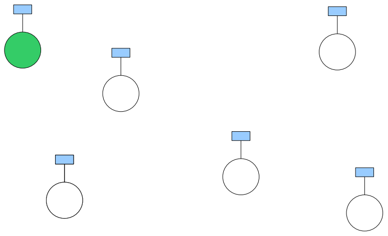
The Contract Net Protocol (CNP) is a widely used task allocation protocol in Multi-Agent Systems. It is fast, flexible and has low communication costs. However it is limited in some issues and has shortcomings if the setting for task assignment is more complicated. Coordination theory is defined as "a body of principles about how activities can be coordinated, and how agents can work together". The task assignment process forms an important part of the coordination and assigning tasks in a multi agent system (MAS) architecture is not simple. When agents have different capabilities and different subproblems require different capabilities, a protocol is needed in order to find the right agent for the right task in an efficient way. The assignment of tasks to agents and the (re-) allocation of tasks in a MAS is one of the key features of automated negotiation systems. One possibility of solving this task assignment problem is using a centralized approach, where all initiators (agents with the tasks) and participants (agents that compete for acquiring tasks) send their information to a central decision maker. This decision maker will decide which participant will perform which task and then notifies it accordingly. This centralized scheme has lower coordination costs, but is very vulnerable to agent failures. Contract Net (CNet) is a well known high-level protocol that uses a decentralized scheme. In CNet, there is no central decision maker. Initiators (also known as managers) are responsible for monitoring the execution of a task and processing the results of its execution. Participants (also known as contractors) are responsible for the actual execution of the tasks. Agents in a MAS are not designated a priori as managers or contractors. These are only roles, and any node can take on either role dynamically during the course of problem solving. The negotiation process of CNet begins by initiators announcing available tasks. Participants evaluate these tasks and submit bids on those for which they are suited. The initiators then evaluate the bids and award contracts to the nodes they determine to be most appropriate. Note that a participant may further partition a task and award contracts to other nodes, by acting as the initiator of that task. Control of the whole task assignment is distributed, because processing and communication are not focused at particular nodes.

7.2 Contract Net Stages



- Recognition;
- Announcement ;
- Bidding;
- Awarding;
- Expediting.

7.2.1 Recognition

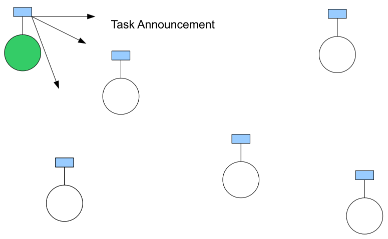


In this stage, an agent recognises it has a problem it wants help with. Agent has a goal, and either

- realises it cannot achieve the goal in isolation - does not have capability;
- realises it would prefer not to achieve the goal in isolation (typically because of solution quality, deadline, etc)

As a result, it needs to involve other agents.

7.2.2 Announcement

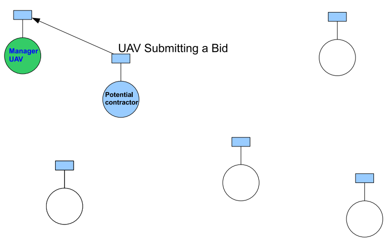


In this stage, the agent with the task sends out an announcement of the task which includes a specification of the task to be achieved.
Specification must encode:

- description of task itself (maybe executable);
- any constraints (e.g., deadlines, quality constraints).
- meta-task information (e.g.,bids must be submitted by...)

The announcement is then broadcast.

7.2.3 Bidding

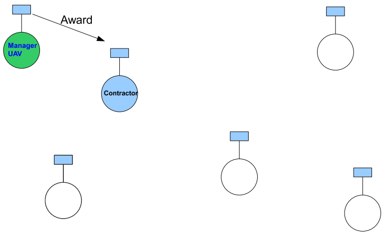


UAVs that receive the announcement decide for themselves whether they wish to bid for the task.
Factors:

- agent must decide whether it is capable of expediting task;
- agent must determine quality constraints and price information (if relevant).

If they do choose to bid, then they submit a tender.

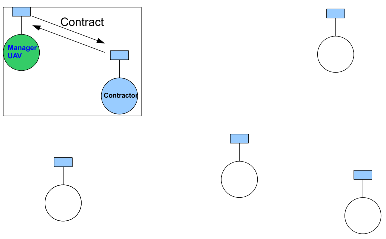
7.2.4 Awarding



Agent that sent task announcement must choose between bids and decide who to "award the contract" to.

The result of this process is communicated to agents that submitted a bid.

7.2.5 Expediting

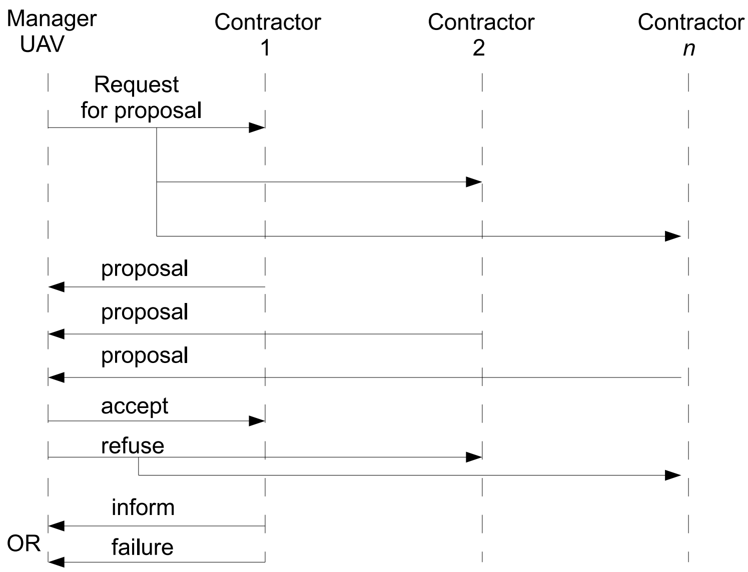


The successful contractor then expedites the task.

May involve generating further manager-contractor relationships: sub-contracting.

Also may involve another contract net protocol.

7.2.6 Procedure Diagram



Issues for Implementing CNP:

- How to specify tasks?
- How to specify quality of service?
- How to decide how to bid?
- How to select between competing offers?
- How to differentiate between offers based on multiple criteria?

7.3 Definition of UAVs and targets

Possible target (task) fields:

- id - task id;
- type -task type;
- value -task reward;
- start-task start time (sec);
- end - task expiry time (sec);
- duration -task default duration (sec);
- x- task position (meters);
- y-task position (meters);
- z-task position (meters).

Possible UAVs fields:

- id- agent id;
- type- agent type;
- avail- agent availability (expected time in sec);
- x- agent position (meters);
- y- agent position (meters);
- z- agent position (meters);
- velocity - agent cruise velocity (m/s));
- fuel-(agent fuel per meter)).

7.4 Agent information

Each agent must carry the foolowing vectors of information in order to be able to perform decentralized algorithm:

N_a Number of agents
N_t- Number of tasks
L_t- Maximum length of the bundle, i.e. each agent can be assigned a maximum L_t tasks
$\mathcal{I}$- Index set of agents where $\mathcal{I} \doteq \{1, ..., N_a\}$
$\mathcal{J}$- Index set of tasks where $\mathcal{J} \doteq \{1, ..., N_t\}$

- A bundle, $\mathbf{b}_i \doteq \{b_{i1}, ..., b_{i|\mathbf{b}_i|}\}$

  of variable length whose elements are defined by $b_{in} \in \mathcal{J}$ for $n = 1, ..., |\mathbf{b}_i|$. The current length of the bundle is denoted by b_i, which cannot exceed the maximum length L_t, and an empty bundle is represented by $b_i = \emptyset$ and $|\mathbf{b}_i| = 0$. The bundle represents the tasks that agent i has selected to do, and is ordered chronologically with respect to when the tasks were added (i.e. task b_{in} was added before task $b_{i(n+1)}$).

- A corresponding path, $\mathbf{p}_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$

  whose elements are defined by $p_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$ for $n = 1, ..., |\mathbf{b}_i|$. The path contains the same tasks as the bundle, and is used to represent the order in which agent i will execute the tasks in its bundle. The path is therefore the same length as the bundle, and is not permitted to be longer than L_t; $|\mathbf{p}_i| = |\mathbf{b}_i| \leq L_t$.

- A vector of times $\tau_i \doteq \{\tau_{i1}, ..., \tau_{i|\tau_i|}\}$

  whose elements are defined by τ_{in} for $n = 1, ..., |\tau_i|$. The times vector represents the corresponding times at which agent i will execute the tasks in its path, and is necessarily the same length as the path.

- A winning agent list $\mathbf{z}_i \doteq \{z_{i1}, ..., z_{iN_t}\}$ of size N_t

  where each element $z_{ij} \in \{\mathcal{I} \cup \emptyset\}$ for $j = 1, ..., N_t$ indicates who agent i believes is the current winner for task j. Specifically, the value in element z_{ij} is the index of the agent who is currently winning task j according to agent i, and is $z_{ij} = \emptyset$; if agent i believes that there is no current winner.

- A winning bid list $\mathbf{y}_i \doteq \{y_{i1}, ..., y_{iN_t}\}$ of size N_t

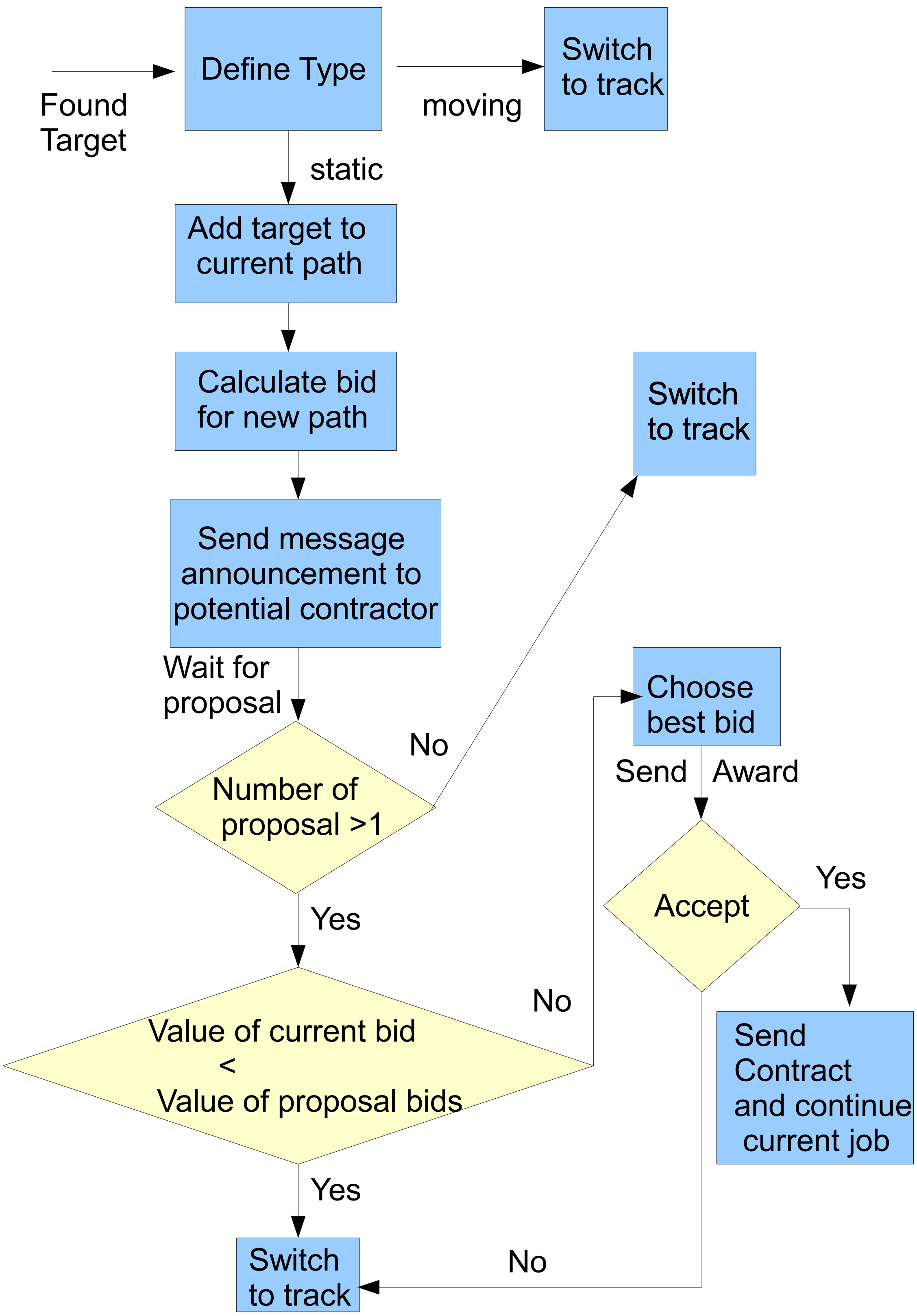
  where the elements $y_{ij} \in [0, \infty)$ represent the corresponding winners bids and take the value of 0 if there is no winner for the task.

Manager UAV_i					$Values$
$Bundle$					$b_i = []$
$Path$					$p_i = []$
$Time$					$\tau_i = []$

$Performing \ \ Search$

Manager UAV_i					$Values$
$Winning \ \ Agent$					$z_i = []$
$WinningBids$					$y_i = []$

7.5 Manager UAVs Logic



For example i-th UAV found a static target;

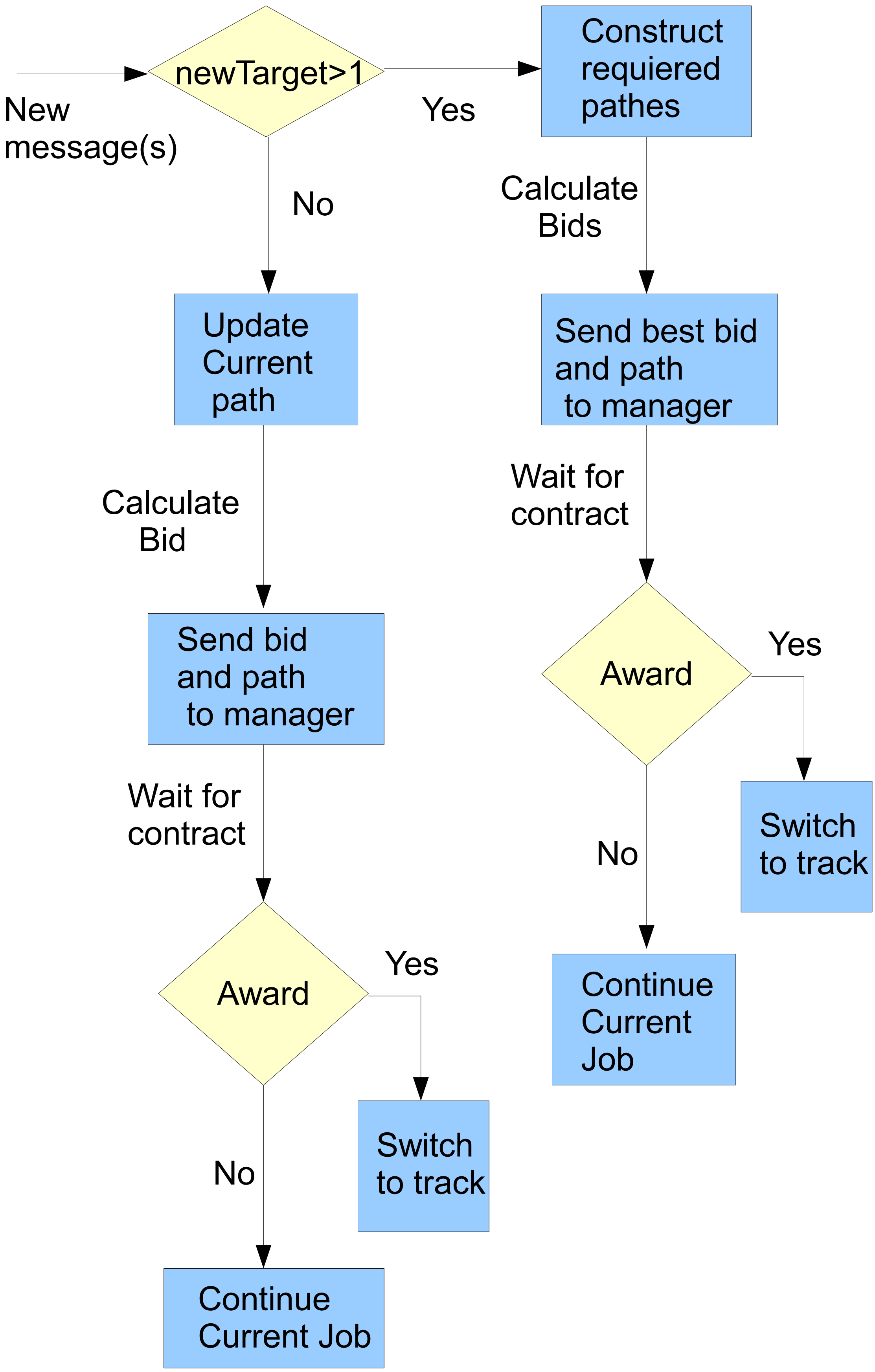
Manager UAV_i	$Target_1$				$Values$
$Bundle$	✠				$b_i = [b_{i1}]$
$Path$	p_{i1}				$p_i = [p_{i1}]$
$Time$	τ_{i1}				$\tau_i = [\tau_{i1}]$

Then this UAV should calculate arrival time $\tau_{i1}(p)$ *and corresponding bid* y_{i1}

Manager UAV_i	$Target_1$				$Values$
$Winning\ \ Agent$	i				$z_i = [z_{i1}]$
$WinningBids$	y_{i1}				$y_i = [y_{i1}]$

And disseminate this information to other UAVs (Potential contractors UAVs), and waiting for their reply.

7.6 Potential Contractor UAVs Logic



For example k-th UAV recieve a message about new target;

UAV_k	$Target_1$	$Target_2$	$Target_k$		$Values$
$Bundle$	$\checkmark$	$\checkmark$	$\checkmark$		$\mathbf{b}_k \leftarrow (\mathbf{b}_k \oplus_{end} 1)$
$Path$	p_{k1}	p_{k2}	p_{kk}		$\mathbf{p}_k \leftarrow (\mathbf{p}_k \oplus_{n_1^*} 1)$
$Time$	τ_{k1}	τ_{k2}	τ_{kk}		$\tau_{\mathbf{k}} \leftarrow (\tau_k \oplus_{n_{1*}} \tau_{k1}(\mathbf{p}_k \oplus_{n_{1*}} 1))$

Update current bundle of targets

Update current pass and arrival time

$\Longrightarrow$ And optimal location n_1^* is then given by $n_1^* = \max_{n_1} c_1(\tau_{k1}^*(\mathbf{p}_k \oplus_{n_1} 1))$

$\Longrightarrow$ Then the final score for new task j(which is include $|b_k|$ targets) is

$$c_{kj}(\mathbf{p}_i) = c_j(\tau_{kj}^*(\mathbf{p}_k \oplus_{n_j^*} j))$$

Then according to the logic above the manager UAVs should compare a bids received from potential contractors UAVs, for example the manager UAV $i = 3$ we can have the following comparison tables:

For case, when bundle of manager UAV3 was not empty $|b_3| \neq \emptyset$

	$Proposal1$	$Proposal2$	$Proposal3$
$UAV1$	c_{11}	-	-
$UAV2$	-	c_{22}	-
$UAV3$	-	-	c_{33}

For case, when bundle of manager UAV3 was empty $|b_3| = \emptyset$

	$Proposal1$	$Proposal2$	$Proposal3$
$UAV1$	c_{11}	-	-
$UAV2$	-	c_{22}	-
$UAV3$	c_{31}	c_{32}	c_{33}

Using Contract Net can be advantageous when compared with other coordination strategies as outlined below.

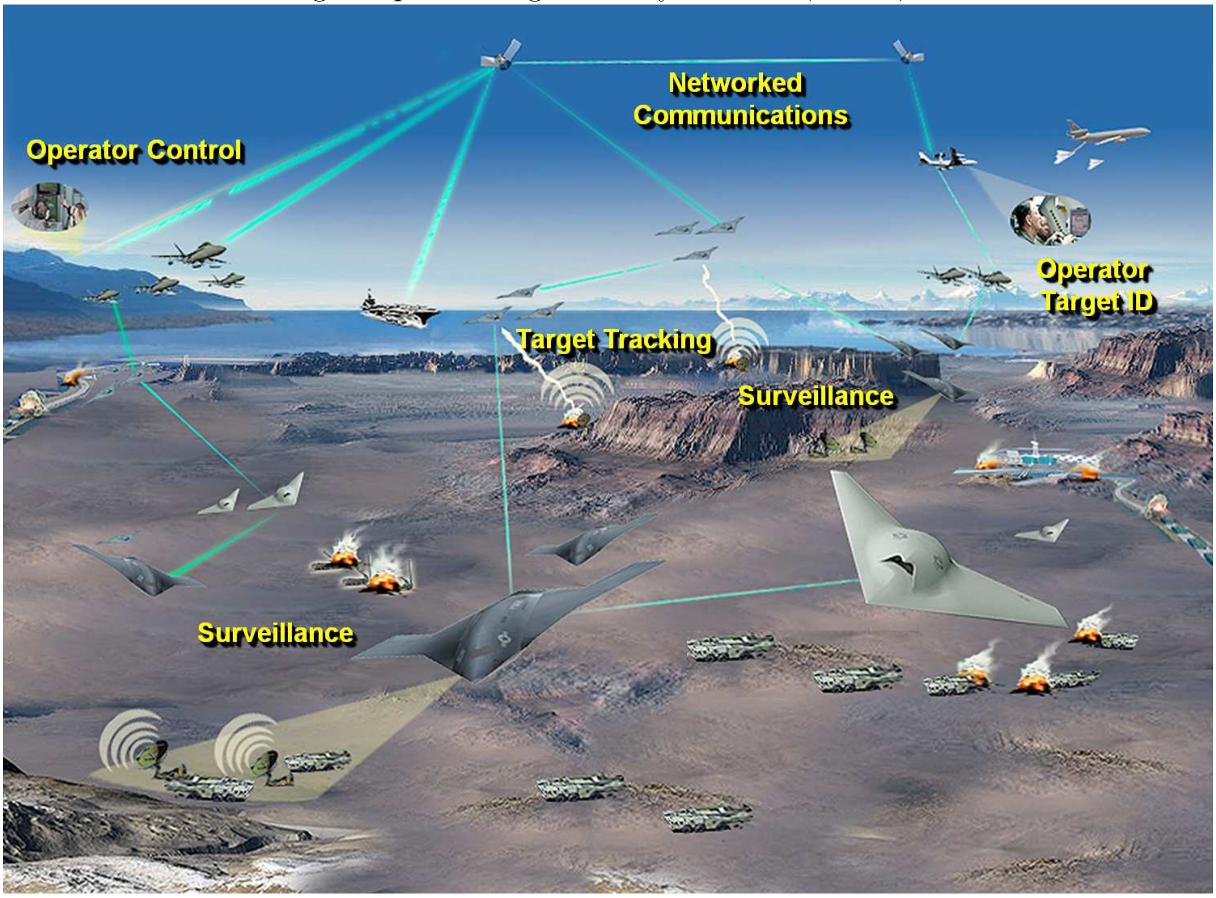
- Tasks are assigned (contracts awarded) dynamically, resulting in the better deals for the parties (agents) involved.
- Agents can enter and leave the system at will.
- The tasks will be naturally balanced among all the agents since agents that already have contract(s) do not have to bid on new ones. If an agent is already using all its resources, it will be unable to bid on new contracts until the current ones are completed.
- A reliable strategy for distributed applications with agents that can recover from failures (to be discussed more in the following paragraphs).

Unfortunately, CNP also have some disadvantages. For example CNP assumes that all agents are friendly and benevolent. As a result, there is no mechanism to detect conflicts and more importantly solve them. By these reason we will introduce the extension of the CNP so-called Consensus-Based Bundle Algorithm (CBBA).

Chapter 8

Consensus-Based Bundle Algorithm

Network centric operations involve large teams of agents, with heterogeneous capabilities, interacting together to perform missions. These missions involve executing several different tasks such as conducting reconnaissance, surveillance, target classification, and rescue operations. Within the heterogeneous team, some specialized agents are better suited to handle certain types of tasks than others. For example, UAVs equipped with video can be used to perform search, surveillance and reconnaissance operations, human operators can be used for classification tasks, ground teams can be deployed to perform rescue operations, etc. Figure below illustrates an example of such complex mission scenario involving numerous networked agents performing a variety of search, track, and surveillance tasks.



Ensuring proper coordination and collaboration between agents in the team is crucial to efficient and successful mission execution, motivating the development of autonomous task allocation methods to improve mission coordination.

Consensus-Based Bundle Algorithm (CBBA) is a decentralized market-based protocol that provides provably good approximate solutions for multi-agent multi-task allocation problems over networks of heterogeneous agents. CBBA consists of iterations between two phases: a bundle building phase where each vehicle greedily generates an ordered bundle of tasks, and a consensus phase where conflicting assignments are identified and resolved through local communication between neighboring agents. There are several core features of CBBA that can be exploited to develop an efficient planning mechanism for heterogeneous teams. First, CBBA is a decentralized decision architecture, which is a necessity for planning over large teams due to the increasing communication and computation overhead required for centralized planning with a large number of agents. Second, CBBA is a polynomial-time algorithm leading to a framework that scales well with the size of the network and/or the number of tasks (or equivalently, the length of the planning horizon). Third, CBBA is capable of handling various design objectives, nonlinear agent models, and constraints, and under a few assumptions on the schoring structure, a provably good feasible solution is guaranteed.

8.1 Problem formulation

The goal of task assignment problem is, given a list of N_t tasks and N_a agents, to find a conflict-free matching of tasks to agents that maximizes some global reward. An assignment is said to be free of conflicts if each task is assigned to no more than one agent. Each agent can be assigned a maximum of L_t tasks, and the assignment is said to be completed once $N_{max} \doteq \min\{N_t, N_a, L_t\}$ tasks have been assigned. The global objective function is assumed to be a sum of local reward values, while each local reward is determined as a function of the tasks assigned to each agent. The task assignment problem described above can be written as the following integer (possibly nonlinear) program with binary decision variables x_{ij} that indicate whether or not task j is assigned to agent i:

$$\sum_{i=1}^{N_a}\left(\sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x_i})))x_{ij}\right) \rightarrow \max \quad (8.1)$$

$$subject \;\; to:$$

$$\sum_{j=1}^{N_t} x_{ij} \leq L_t, \quad \forall i \in \mathcal{I}$$

$$\sum_{i=1}^{N_a} x_{ij} \leq 1, \quad \forall j \in \mathcal{J}$$

$$\sum_{i=1}^{N_a}\sum_{j=1}^{N_t} = N_{max}$$

$$x_{ij} \in \{0,1\}, \forall (i,j) \in \mathcal{I} \times \mathcal{J}$$

where $x_{ij} = 1$ if agent i is assigned to task j, and $\mathbf{x}_i \doteq \{x_{i1}, ..., x_{iN_t}\}$ is a vector of assignments for agent i, whose j-th element is x_{ij}.

The summation term in brackets in the objective function represents the local reward for agent i.

N_a Number of agents
N_t- Number of tasks
L_t- Maximum length of the bundle, i.e. each agent can be assigned a maximum L_t tasks
$\mathcal{I}$- Index set of agents where $\mathcal{I} \doteq \{1, ..., N_a\}$
$\mathcal{J}$- Index set of tasks where $\mathcal{J} \doteq \{1, ..., N_t\}$
$\mathbf{p}_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$ - The variable length vector represent the path for agent i,an ordered sequence of tasks where the elements are the task indices, $p_{in} \in \mathcal{J}$ for $n = 1, ..., |\mathbf{p}_i|$, i.e. its n-th element is $j \in \mathcal{J}$ if agent i conducts task j at the n-th point along the path. The current length of the path is denoted by $|\mathbf{p}_i| \leq L_t$.

Also note that score function $c_{ij}(\mathbf{x}_i, \mathbf{p}_i)$ is assumed to be nonnegative. The score function can be any nonnegative function of either assignment $\mathbf{x}_i$ or path $\mathbf{p}_i$ (usually not a function of both); in the context of task assignment for unmanned vehicles with mobility, it often represents a path-dependent reward such as the path length, the mission completion time, and the time-discounted value of target.

8.2 Key assumptions

- The score c_{ij} that agent i obtains by performing task j is defined as a function of the arrival time τ_{ij} at which the agent executes the task (or possibly the expected arrival time in a probabilistic setting).
- The arrival time τ_{ij} is uniquely defined as a function of the path $\mathbf{p}_i$ that agent i takes.

- The path $\mathbf{p}_i$ is uniquely defined by the assignment vector of agent $i, \mathbf{x}_i$.

Several design objectives commonly used for multi-agent decision making problems feature scoring functions that satisfy the above set of assumptions. An example is the problem involving time-discounted values of targets, in which the sooner an agent arrives at the target, the higher the reward it obtains. Or for scenario involves re-visit tasks, where previously observed targets must be revisited at some scheduled time. In this case the score function would have its maximum at the desired re-visiting time and lower values at other re-visit times.

8.3 Vectors of agent information

The scoring function in (8.1) depends on the assignment vector $\mathbf{x}_i$ and on the path $\mathbf{p}_i$, which makes this integer programming problem significantly complex for $L_t > 1$. To address these dependencies, the classical combinatorial auction methods explored simplifications of the problem by treating each assignment combination or "bundle" of tasks as a single item for bidding. These approaches led to complicated winner selection methods as well as bad scalability due to the increase in computation associated with enumerating all possible bundle combinations. In contrast, the Consensus-Based Bundle Algorithm (CBBA) consists of an auction process which is performed at the task level rather than at the bundle level, where agents build their bundles in a sequential greedy fashion.

And in order to perform this decentralized algorithm each agent must carry the following vectors of information:

- A bundle, $\mathbf{b}_i \doteq \{b_{i1}, ..., b_{i|\mathbf{b}_i|}\}$

  of variable length whose elements are defined by $b_{in} \in \mathcal{J}$ for $n = 1, ..., |\mathbf{b}_i|$. The current length of the bundle is denoted by b_i, which cannot exceed the maximum length L_t, and an empty bundle is represented by $b_i = \emptyset$ and $|\mathbf{b}_i| = 0$. The bundle represents the tasks that agent i has selected to do, and is ordered chronologically with respect to when the tasks were added (i.e. task b_{in} was added before task $b_{i(n+1)}$).

- A corresponding path, $\mathbf{p}_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$

  whose elements are defined by $p_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$ for $n = 1, ..., |\mathbf{b}_i|$. The path contains the same tasks as the bundle, and is used to represent the order in which agent i will execute the tasks in its bundle. The path is therefore the same length as the bundle, and is not permitted to be longer than L_t; $|\mathbf{p}_i| = |\mathbf{b}_i| \leq L_t$.

- A vector of times $\tau_i \doteq \{\tau_{i1}, ..., \tau_{i|\tau_i|}\}$

  whose elements are defined by τ_{in} for $n = 1, ..., |\tau_i|$. The times vector represents the corresponding times at which agent i will execute the tasks in its path, and is necessarily the same length as the path.

- A winning agent list $\mathbf{z}_i \doteq \{z_{i1}, ..., z_{iN_t}\}$ of size N_t

  where each element $z_{ij} \in \{\mathcal{I} \cup \emptyset\}$ for $j = 1, ..., N_t$ indicates who agent i believes is the current winner for task j. Specifically, the value in element z_{ij} is the index of the agent who is currently winning task j according to agent i, and is $z_{ij} = \emptyset$; if agent i believes that there is no current winner.

- A winning bid list $\mathbf{y}_i \doteq \{y_{i1}, ..., y_{iN_t}\}$ of size N_t

  where the elements $y_{ij} \in [0, \infty)$ represent the corresponding winners bids and take the value of 0 if there is no winner for the task.

- Vector of timestamps $\mathbf{s}_i \doteq \{s_{i1}, ..., s_{iN_a}\}$, of size N_a

  where each element $s_{ik} \in [0, \infty)$ for $k = 1, ..., N_a$ represents the timestamp of the last information update agent i received about agent k, either directly or through a neighboring agent.

$$\Downarrow$$

Each agent must carry these vectors of information in order to be able to perform decentralized algorithm which consists of iterations between two phases:
a bundle building phase where each vehicle greedily generates an ordered bundle of tasks, and a
consensus phase where conflicting assignments are identified and resolved through local communication between neighboring agents.
Algorithm will iterates between these two phases until no changes to the information vectors occur anymore.

8.4 Bundle construction phase

In contrast to the combinatorial algorithms, which enumerate all possible bundles for bidding, in CBBA each agent creates just its single bundle which is updated as the assignment process progresses. During this phase of the algorithm, each agent continuously adds tasks to its bundle in a sequential greedy fashion until it is incapable of adding any others. Tasks in the bundle are ordered based on which ones were added first in sequence, while those in the path are ordered based on their predicted execution times.

The bundle construction process is as follows: for each available task not currently in the bundle, or equivalently not in the path ($j \notin \mathbf{p}_i$), the agent computes a score for the task, $c_{ij}(\mathbf{p}_i)$. The score is checked against the current winning bids, and is kept if it is greater. Out of the remaining ones, the agent selects the task with the highest score and adds that task to its bundle.

Computing the score for a task is a complex process which is dependent on the tasks already in the agents path (and/or bundle). Selecting the best score for task j can be performed using the following two steps. First, task j is "inserted" ($\oplus_{n_j}$)in the path at some location n_j (the new path becomes ($\mathbf{p}_i \oplus_{n_j} j$), where $\oplus_n$ signifies inserting the task at location n). The score for each task $c_j(\tau)$ is dependent on the time at which it is executed, motivating the second step, which consists of finding the optimal execution time given the new path, $\tau^*_{ij}(\mathbf{p}_i) \oplus_{n_j} j$. This can be found by solving the following optimization problem:

$$\begin{aligned} \tau^*_{ij}(\mathbf{p}_i \oplus_{n_j} j) = & \max_{\tau_{ij} \in [0,\infty)} c_j(\tau_{ij}) \\ & subject \;\; to: \\ \tau^*_{ik}(\mathbf{p}_i \oplus_{n_j} j) = & \; \tau^*_{ik}, \forall k \in \mathbf{p}_i \end{aligned} \tag{8.2}$$

The constraints state that the insertion of the new task j into path $\mathbf{p}_i$ cannot impact the current times (and corresponding scores) for the tasks already in the path. Note that this is a continuous time optimization, which, for the general case, involves a significant amount of computation. The optimal score associated with inserting the task at location n_j is then given by $c_j(\tau^*_{ij}(\mathbf{p} \oplus_{n_j} j))$. This process is repeated for all n_j by inserting task j at every possible location in the path. The optimal location is then given by

$$n^*_j = \max_{n_j} c_j(\tau^*_{ik}(\mathbf{p}_i \oplus_{n_j} j)) \tag{8.3}$$

and the final score for task j is $c_{ij}(\mathbf{p}_i) = c_j(\tau^*_{ij}(\mathbf{p}_i \oplus_{n^*_j} j))$.

Once the scores for all possible tasks are computed ($c_{ij}(\mathbf{p}_i)\forall j \notin \mathbf{p}_i$)), the scores need to be checked against the winning bid list, y_i, to see if any other agent has a higher bid for the task. We define the variable $h_{ij} = I(c_{ij}(\mathbf{p}_i) > y_{ij})$, where $I(\cdot)$ denotes the indicator function that equals unity if the argument is true and zero if it is false, so that $c_{ij}(\mathbf{p}_i)h_{ij}$ will be nonzero only for viable bids. The final step is to select the highest scoring task to add to the bundle:

$$j^* = \max_{j \notin \mathbf{p}_i} c_{ij}(\mathbf{p}_i) h_{ij} \tag{8.4}$$

The bundle, path, times, winning agents and winning bids vectors are then updated to include the new task:

$$\begin{aligned} \mathbf{b}_i &\leftarrow (\mathbf{b}_i \oplus_{end} j^*) \\ \mathbf{p}_i &\leftarrow (\mathbf{p}_i \oplus_{n_{j^*}} j^*) \\ \tau_i \leftarrow (\mathbf{tau}_i \oplus_{n_{j^*}} \tau^*_{ij^*}(\mathbf{p}_i &\oplus_{n_{j^*}} j^*)) \\ z_{ij} &= i \\ y_{ij} &= c_{ij^*}(\mathbf{p}_i) \end{aligned} \tag{8.5}$$

The bundle building recursion continues until either the bundle is full (the limit L_t is reached), or no tasks can be added for which the agent is not outbid by some other agent ($h_{ij} = 0 \forall j \notin \mathbf{p}_i$). Notice that with equation (8.5), a path is uniquely defined for a given bundle, while multiple bundles might result in the same path.

8.5 Consensus phase

Once agents have built their bundles of desired tasks they need to communicate with each other to resolve conflicting assignments amongst the team. After receiving information from neighboring agents about the winning agents and corresponding winning bids, each agent can determine if it has been outbid for any task in its bundle. Since the bundle building recursion, described in the previous section, depends at each iteration upon the tasks in the bundle up to that point, if an agent is outbid for a task, it must release it and all subsequent tasks from its bundle. If the subsequent tasks are not released, then the current best scores computed for those tasks would be overly conservative, possibly leading to a degradation in performance. It is better, therefore, to release all tasks after the outbid task and redo the bundle building recursion process to add these tasks (or possibly better ones) back into the bundle.

This consensus phase assumes that each pair of neighboring agents synchronously shares the following information vectors: the winning agent list z_i, the winning bids list y_i, and the vector of timestamps s_i representing the time stamps of the last information updates received about all the other agents. The timestamp vector for any agent i is updated using the following equation,

$$s_{ik} = \begin{cases} \tau_r(i.e.\ message\ reception\ time), & if\ g_{ik} = 1; \\ \max\{s_{mk} | m \in \mathcal{I}, g_{im} = 1\}, & otherwise \end{cases} \tag{8.6}$$

which states that the timestamp s_{ik} that agent i has about agent k is equal to the message reception time τ_r if there is a direct link between agents i and k (i.e. $g_{ik} = 1$ in the network graph), and is otherwise determined by taking the latest timestamp about agent k from the set of agent i's neighboring agents. For each message that is passed between a sender k and a receiver i, a set of actions is executed by agent i to update its information vectors using the received information. These actions involve comparing its vectors z_i, y_i, and s_i to those of agent k to determine which agent's information is the most up-to-date for each task. There are three possible actions that agent i can take for each task j:

i, (receiver)	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$					$z_i = [z_{i1}, z_{i2}, ...]$
$WinningBids$					$y_i = [y_{i1}, y_{i2}, ...]$
$Timesstamps$					$s_i = [s_{i1}, s_{i2}, ...]$

$$Update: \quad z_{ij} = z_{kj}, \quad y_{ij} = y_{kj}$$

$$Reset: \quad z_{ij} = \emptyset, \quad y_{ij} = 0$$

$$Leave: \quad z_{ij} = z_{ij}, \quad y_{ij} = y_{ij}$$

k, (sender)	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$					$z_k = [z_{k1}, z_{k2}, ...]$
$WinningBids$					$y_k = [y_{k1}, y_{k2}, ...]$
$Timesstamps$					$s_k = [s_{k1}, s_{k2}, ...]$

Next we will give the decision rules for this communication protocol.

8.5.1 Decision Rules

Agent k thinks z_{kj} is	Agent i thinks z_{ij} is	Receiver Action
k	i	if $y_{kj} > y_{ij} \rightarrow update$
k	k	$update$
k	$m \notin \{i,k\}$	$if\ s_{km} > s_{im}$ or $y_{kj} > y_{ij} \rightarrow update$
k	$none$	$update$

$$s_{ik} = \begin{cases} \tau_r(i.e.\ message\ reception\ time), & if\ g_{ik} = 1; \\ \max\{s_{mk} \mid m \in \mathcal{I}, g_{im} = 1\}, & otherwise \end{cases}$$

Agent k thinks z_{kj} is	Agent i thinks z_{ij} is	Receiver Action
i	i	$leave$
i	k	$reset$
i	$m \notin \{i,k\}$	$if\ s_{km} > s_{im} \rightarrow reset$
i	$none$	$leave$

The first two columns of the table indicate the agent that each of the sender k and receiver i believes to be the current winner for a given task; the third column indicates the action that the receiver should take, where the default action is "Leave".

Agent k thinks z_{kj} is	Agent i thinks z_{ij} is	Receiver Action
$m \notin \{i,k\}$	i	$if\ s_{km} > s_{im}\ and\ y_{kj} > y_{ij} \rightarrow update$
$m \notin \{i,k\}$	k	$if\ s_{km} > s_{im} \rightarrow update$ $else \rightarrow reset$
$m \notin \{i,k\}$	m	$s_{km} > s_{im} \rightarrow update$
$m \notin \{i,k\}$	$n \notin \{i,k,m\}$	$if\ s_{km} > s_{im}\ and\ s_{kn} > s_{in} \rightarrow update$ $if\ s_{km} > s_{im}\ and\ y_{kj} > y_{ij} \rightarrow update$ $if\ s_{kn} > s_{in}\ and\ s_{im} > s_{km} \rightarrow reset$
$m \notin \{i,k\}$	$none$	$if\ s_{km} > s_{im} \rightarrow update$

$$s_{ik} = \begin{cases} \tau_r, & if\ g_{ik} = 1; \\ \max\{s_{mk} \mid m \in \mathcal{I}, g_{im} = 1\}, & otherwise \end{cases}$$

Agent k thinks z_{kj} is	Agent i thinks z_{ij} is	Receiver Action
$none$	i	$leave$
$none$	k	$update$
$none$	$m \notin \{i,k\}$	$if\ s_{km} > s_{im} \rightarrow update$
$none$	$none$	$leave$

If either of the winning agent or winning bid information vectors (z_i or y_i) are changed as an outcome of the communication, the agent must check if any of the updated or reset tasks were in its bundle. If so, those tasks, along with all others added to the bundle after them, are released. Thus if $\overline{n}$ is the location of the first outbid task in the bundle ($\overline{n} = \min\{n | z_{i(b_{in})} \neq i\}$ with b_{in} denoting the n-th entry of the bundle), then for all bundle locations $n \geq \overline{n}$, with corresponding task indices bin, the following updates are made:

$$\begin{aligned} z_{i(b_{in})} &= \\ y_{i(b_{in})} &= 0 \end{aligned} \tag{8.7}$$

The bundle is then truncated to remove these tasks,

$$b_i \leftarrow \{b_{i1}, ..., b_{i(\overline{n}-1)}\} \tag{8.8}$$

and corresponding entries are removed from the path and times vectors as well. From here, the algorithm returns to the first phase where new tasks can be added to the bundle. CBBA iterates between these two phases until no changes to the information vectors occur anymore.

8.6 Algorithm summary

- Calculate marginal score for all tasks

$$c_{ij}(\mathbf{p}_i) = \begin{cases} 0, & if\ j \in \mathbf{p}_i; \\ \max_{n \leq l_b} S_{path}(\mathbf{p}_i \oplus_n j) - S_{path}(\mathbf{p}_i), & otherwise \end{cases}$$

- Determine which tasks are winnable

  $h_{ij} = \mathbf{I}(c_{ij}(\mathbf{p}_i) > y_{ij}), \forall j \in \mathcal{J}$

- Select the index of the best eligible task, j^*, and select best location in the plan to insert the task, n_j^*

$$j^* = \max_{j \in \mathcal{J}} c_{ij} h_{ij}$$

$$n_j^* = \max_{n \in \{0,\ldots,l_b\}} S_{path}(\mathbf{p}_i \oplus_n j^*)$$

- If $c_{ij^*} \leq 0$, then return. otherwise, continue

- Update agent information

$$\mathbf{b}_i \leftarrow (\mathbf{b}_i \oplus_{l_b} j^*)$$

$$\mathbf{p}_i \leftarrow (\mathbf{p}_i \oplus_{n_j^*} j^*)$$

- Update shared information vectors

$$y_{i(j^*)} = c_{i(j^*)}$$

$$z_{i(j^*)} = i$$

- if $l_b = L_t$, then return, otherwise, go to 1.

8.7 Simulation result

The current version of CBBA supports tasks with time windows of validity, homogenous agent-task compatibility requirements, and score functions that balance task reward and fuel costs. Namely to the given local information the algorithm allows the UAVs manage themselves automatically as a team in a distributed manner. The centralized agency is no longer involved in receiving requests and allocating UAVs to the requests. It only needs to ensure that there are sufficient UAVs in the pool to service the requests.

The general system configuration for simulation research of distributed tasking problem are as follows

Configuration:

- UAV Airspace;
- Range for air-to-air communications between the UAVs;
- Homogenous Service UAVs;
- Local information about targets;

- Static and moving targets.

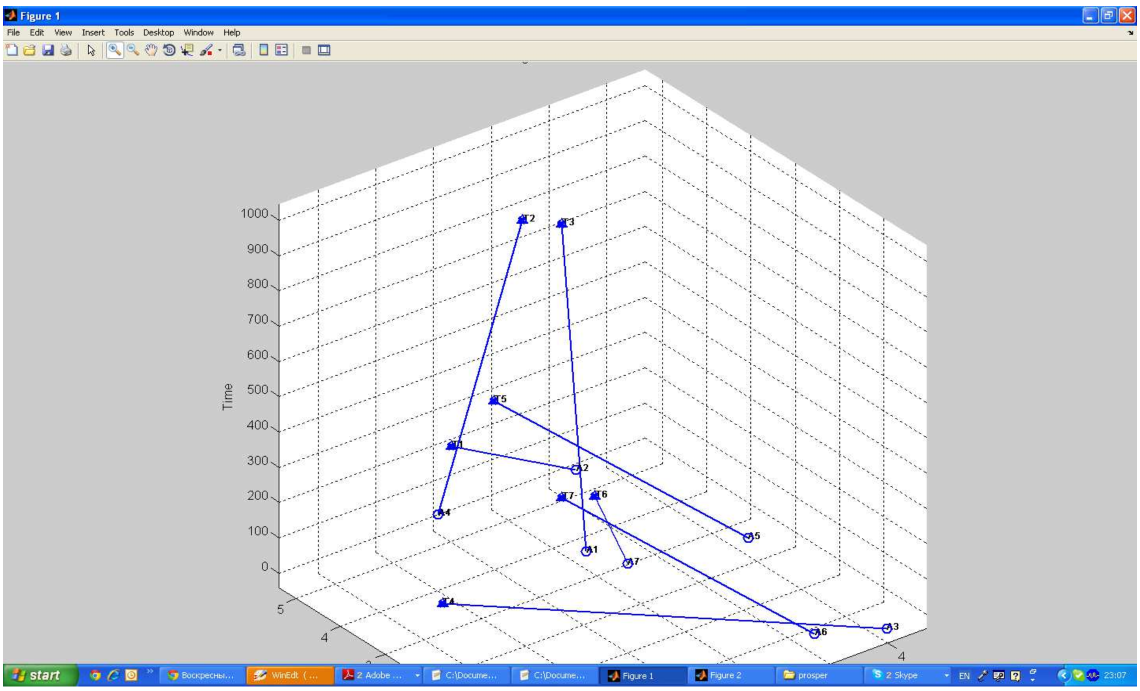
Requirements:

- Manage UAVs themselves automatically as a team in a hierarchical and / or distributed manner;
- Provide automatic self-allocation and self-deployment.;
- Maximize the local reward for each Service-UAVs.

Considerations:

- Limited UAV sensing range and communication range;
- Breakdown of UAVs and variations of communication topologies;
- Distributed information fusion between UAVs;
- Distributed motion planning for UAVs;

For example the so called Tethered UAVs Self-Assignment Problem is a particular case of the problem described above. And the solution of this particular problem presented in a figure below, namely, we find the logic that enabled the tethered UAVs (7 UAVs) to self-deploy one UAV to each specified location (7 locations).



Chapter 9

Conclusion and Further work

We have presented a method for using a Integer Linear Program (ILP) formulation to find the optimal solution to multiple-task assignment problem where the tasks are coupled by timing and other constraints. This formulation allows variation of UAVs flight paths to guarantee that timing constraints are satisfied, and directly incorporates the varying task completion times into the optimization. This is a promising formulation, which allows a true optimal solution for a vary challenging problem. Solution results were presented for practical problem sizes, but scaling issues will require further work before the method can be applied to large problems. Future work will simply the problem structure to reduce complexity and apply the method to task assignment problem in a detailed UAV simulation, including more realistic cost functions.

For design the effective numerical realization of a dynamical tasking problem we develop the new optimality and sub-optimality conditions that are more suitable for the design of numerical methods and further applications. In contrast to the classic approaches, we proposed uses the idea of constructive approach and extend this setting to produce new results and constructive elements of optimization theory for the considered MAS systems and state also its relevant basic properties which can be of interest for others purposes, too. It is expected that the obtained optimality and sub-optimality conditions will be close related to the corresponding classic results of maximum principle and "epsilon"- maximum principle. Optimal control is exploited usually as an effective and perspective way to improve the desired system performance and characteristics. In practical sense the optimal feedback control low is more reasonable. With this motivation, the major advantage of the proposed constructive approach is that the sensitivity analysis and some differential properties of the optimal controls under disturbances can be studied which is very important for their application to the optimal synthesis problem. It has been conjectured that such setting could be appropriate for development of numerical methods also.

Bibliography

[1] L. F. Bertuccelli, W.M. Beckers, M.L Cummings: Developing operator models for UAVs search scheduling, AIAA Guidance, Navigation, and Control Conference 2-5 August, 2010, Toronto, Ontario, Canada

[2] Gabasov R., Kirillova F. M. and Prischepova S. V.: Optimal Feedback Control, 1995, Springer-Verlag Lecture Notes in Control and Information Sciences Series, vol. 207.

[3] R. Gabasov, F.M. Kirillova, O.I. Kostyukova, Constructive methods of optimization, Part 3(Network problems) University Press, Minsk 1986 (in Russian)

[4] Danzig G.B, Ford L.R, Fulkerson D.R. A primal-dual algorithm for linear programming. In: Kuhn HW, Tucker AW, editors. Linear inequalities and related systems. New Jersey: Princeton University Press; 1956, p. 171-81.

[5] S. Dymkou, Graph and 2-D Optimization Theory and their application for discrete simulation of gas transportation networks and industrial processes with repetitive operations. PhD Thesis, RWTH, Aachen, Germany 2006.

[6] Adam N. Elmachtoub, Charles F. van Loan, :From Random Polygon to Ellipse: An Eigenanalysis, SIAM Review, 2010, Vol.52, No. 1, pp.151-170

[7] Kostyukova O. I.: Parametric optimal control problem with variable index, 2003, Computational Mathematics and Mathematical Physics, 43(1), pp. 26–41 (translated from Journal Vychislitelnoi Mathematiki i Mathematicheskoi Fiziki, 2003, v. 43, No. 1, pp. 26–41.

[8] A. Martin and M. Moeller. Mixed integer models for the stationary case of gas network optimization. Technical report. TU Darmstadt. 2004.

[9] Ford L.R, Fulkerson D.R. Flows in networks. New Jersey: Princeton University Press; 1962.

[10] Wei Ren, Yongcan Cao. Distributed Coordination of Multi-agent Networks. Springer-Verlag London Limited, 2011