

EÖTVÖS LORÁND TUDOMÁNYEGYETEM
TERMÉSZETTUDOMÁNYI KAR

**OPTIMALIZÁLÁSI MODELLEK TENGERI SZÉLFARMOK
KARBANTARTÁSÁHOZ**

BSc Szakdolgozat

Készítette: Végh Cintia Kinga
Matematika BSc, Matematikai elemző szakirány

Témavezető: Bérczi-Kovács Erika
Operációkutatási Tanszék



Budapest, 2017

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani témavezetőmnek, Bérczi-Kovács Erikának a témáért, a rengeteg segítségért, magyarázatért és tanácsért. Hálás vagyok továbbá mindazért a türelemért, amivel mindvégig fordult felém.

Emellett köszönettel tartozok a családomnak a rengeteg támogatásért, biztatásért, szeretetért és belém vetett hitükért.

Tartalomjegyzék

1. Bevezetés	4
Bevezetés	4
2. Matematikai háttér	6
2.1. A dualitás	6
2.2. Többtermékes folyamok	8
2.2.1. A Ford-Fulkerson oszlopgenerálás algoritmus (1958)	8
2.3. A Dantzig-Wolfe dekompozíció	11
3. Optimalizálási modellek tengeri szélfarmok karbantartásának ütemezéséhez	16
3.1. Bevezetés	16
3.2. Kihívások	17
3.3. A feladat és megoldásai	18
3.3.1. Első lépések egy hosszú úton	19
3.3.2. Innovatív módszer a megoldásban	23
3.3.3. Egy újabb mérföldkő napjainkból	24
3.4. Futási eredmények	30
3.5. Konklúzió és további lehetőségek	32

1. fejezet

Bevezetés

Szakedolgozatom témájának az optimalizálást választottam. Azt hiszem, joggal mondhatom, hogy régóta mozgatja a világot az a gondolat, hogy hogyan lehet bizonyos szempontok alapján a lehető legjobbat kihozni például egy döntési helyzetből, egy állapotból vagy egyéb szituációból. Az optimalizálás során tökéletességre, hatékonyságra, az igények és szükségletek teljes körű kielégítésére törekszünk. Legyen szó költségminimalizálásról vagy épp ellenkezőleg profitmaximalizálásról, netán útvonaltervezésről, a termelés ütemezéséről, vagy egészen hétköznapi dolgokról, a háztartásoktól kezdve a nagyvállalatokig, mindenütt jelen van az életünkben. Ennek ellenére a legtöbb esetben cseppet sem mondható triviális problémának, sőt, egészen komplex feladatot jelenthet azoknak, akik optimalizálásra adják a fejüket. Szerencsére a matematika eszközei megkönnyítik a helyzetünket, segítségükkel ugyanis felírhatunk úgynevezett matematikai modelleket, melyek képesek bennünket hozzásegíteni a keresett optimális megoldásokhoz.

A példák során említhettem volna az energia iránti kereslet optimalizálását is, hiszen nem elhanyagolható szerepet tölt be életünkben, arról nem is beszélve, hogy szoros kapcsolatban áll a felsoroltakkal. A megújuló energiaforrások térnyerésével egyre nagyobb szükség van az innovatív megoldásokra, a kreatív ötletekre, és a folyamatos fejlődésre. A szélenergia terén elért sikerek rendkívül figyelemre méltóak, nem véletlen tehát, hogy a kereslete is egyre nagyobb méreteket ölt. Ennek okán és további előnyös tulajdonságai miatt (például, hogy a szélenergia felhasználásával a szennyezőanyag-kibocsátás csökkenthető), a termelése rohamos iramban növekszik, amely együtt jár a szélerőművek és szélfarmok terjedésével, ami egészen a gondolatuk felvetésétől kezdve, a telepítésükön át, a karbantartásukig újabb és újabb optimalizációs problémákat vet fel.

Dolgozatom néhány, a tengeri szélfarmok karbantartásának ütemezéséhez kapcsolódó optimalizálási modell bemutatásáról szól majd, melyekhez a szükséges matematikai háttér munkám második fejezetében olvasható. Ebben a részben először a lineáris programozás és a dualitás témakörének alapfogalmai, néhány fontosabb állítás, és a dualitás tétel speciális alakja szerepel majd. Ezután definiálom a többtermékes folyamatokat, amikhez kapcsolódóan ismertetem a Ford-Fulkerson-féle oszlopgenerálás algoritmust, majd kitérek egy nagyon érdekes és összetett módszerre, a Dantzig-Wolfe dekompozícióra. Ezek az ismeretek rendkívül hasznosak lesznek a harmadik fejezetben leírtak olvasásakor. Ebben a részben szó esik többek között korábbi eredményekről, a kihívásokról, a problémakörrel és a feladatról részletesen, valamint három izgalmas modell megoldását is ismertetem, melyek jól igazolják a fent leírtakat, vagyis hogy az új

ötletek elengedhetetlenek a fejlődéshez. Mivel a modellek egyre nagyobb méreteket öltenek, szükség volt olyan módszerekre, amivel a megoldásuk megkönnyíthető. Az utolsó két modellben erre a célra a már említett Dantzig-Wolfe dekompozíciót alkalmazták, méghozzá sikeresen, amit a fejezet végén leírt futási eredmények is igazolnak. Végül zárásként további lehetőségekről is szó lesz majd.

2. fejezet

Matematikai háttér

Ebben a fejezetben azokat a matematikai fogalmakat, tételeket és módszereket gyűjtöttem össze, amelyekre a tengeri szélfarmok karbantartásának ütemezéséhez kapcsolódó optimalizálási modellek megértéséhez lesz szükség. Szó esik a dualitás témakörének néhány alapvető tudnivalójáról, és a Ford-Fulkerson szerzőpáros többtermékes folyamatokhoz kapcsolódó oszlopgenerálás algoritmusáról is. Végül, de nem utolsó sorban a Dantzig-Wolfe dekompozíciós módszer rejtelmeibe nyerhet betekintést az olvasó.

2.1. A dualitás

Az optimalizálási problémákat gyakran a lineáris programozás segítségével oldjuk meg, ami lineáris egyenlőtlenség-rendszerek megoldásával foglalkozik, általában egy lineáris függvény szélsőértékének megtalálása érdekében. A feladatok többsége normál alakban adott:

$$Ax \leq b$$

$$x \geq 0$$

$$\max c^T x$$

ahol $A \in \mathbb{R}^{m \times n}$ ($m \leq n$) az együttható mátrix, $b \in \mathbb{R}^m$ a kapacitásvektor, $c^T \in \mathbb{R}^n$ vektor tartalmazza a célfüggvény együtthatóit, míg $x \in \mathbb{R}^n$ a változók vektora. Sokszor láthatunk kanonikus alakú lineáris programozási feladatot (LP feladatot) is, amelyben $Ax = b$, $x \geq 0$, $\max c^T x$. A normál alakú feladatok is könnyen átalakíthatóak ilyenné m darab új, úgynevezett eltérésváltozó bevezetésével. Ha z az eltérésváltozók vektora, akkor az $Ax + Iz = b$, $x \geq 0$, $\max c^T x$ feladat máris kanonikus alakú, és optimális megoldásai megegyeznek a normál alakú LP feladat optimális megoldásaival. A következő definíciók és tételek a kanonikus alakú feladatokra vonatkoznak majd, ahol feltesszük, hogy A sorai lineárisan függetlenek.

2.1.1. Definíció (Megoldás). Azt az $x \in \mathbb{R}^n$ vektort, amely eleget tesz az $Ax = b$ feltételnek, megoldásnak nevezzük.

2.1.2. Definíció (Megengedett megoldás). Azt mondjuk, hogy az $x \in \mathbb{R}^n$ vektor megengedett megoldás, ha az $Ax = b$ és az $x \geq 0$ feltételeket is kielégíti.

2.1.3. Definíció (Optimális megoldás). Az $x \in \mathbb{R}^n$ vektort optimális megoldásnak nevezzük, ha x megengedett megoldás, és bármely x' megengedett megoldás esetén $cx \geq cx'$.

2.1.4. Definíció (Bázis). Bázisnak nevezünk egy olyan \mathbf{B} , a sorok halmazából az oszlopok halmazába történő leképezést, amelyre $a_{\mathbf{B}_1}, a_{\mathbf{B}_2}, \dots, a_{\mathbf{B}_m}$ oszlopvektorok lineárisan függetlenek.

Jelöljük $x_{\mathbf{B}}$ -vel az x vektor, $c_{\mathbf{B}}$ -vel pedig a c vektor \mathbf{B} bázishoz tartozó részét.

2.1.5. Definíció (B bázishoz tartozó bázismegoldás). Az $x \in \mathbb{R}^n$ vektort a \mathbf{B} bázishoz tartozó bázismegoldásnak nevezzük, ha a bázison kívüli komponensei nullák, a bázishoz tartozó komponenseire pedig teljesül, hogy $Bx_{\mathbf{B}} = b$, ahol B a \mathbf{B} bázishoz tartozó bázismátrix. Ilyen megoldás létezik és egyértelmű: $x_{\mathbf{B}} = B^{-1}b$. (A B bázismátrix invertálható, mert teljes rangú.)

2.1.6. Definíció (Megengedett bázis). A \mathbf{B} bázist megengedett bázisnak nevezzük, ha a hozzá tartozó bázismegoldás megengedett, azaz $x_{\mathbf{B}} = B^{-1}b \geq 0$.

A lineáris programozási feladatok megoldásához általában a szimplex módszert hívjuk segítségül, melynek során sokkal könnyebb úgy dolgozni, ha az $Ax = b$ egyenletrendszert olyan alakba transzformáljuk, hogy a bázismátrix helyén egy egységmátrix legyen. Ehhez meg kell szorozni az egyenletrendszert balról B^{-1} -zel, azaz a transzformált mátrixunk $\bar{A} = B^{-1}A$, a jobb oldalunk pedig $\bar{b} = B^{-1}b$ alakú lesz. A módszer során a redukált költség előjele dönti el, hogy tovább kell-e számolnunk vagy sem. De mi is a redukált költség?

2.1.7. Definíció (Redukált költség). A \mathbf{B} bázishoz tartozó redukált költség a következő: $\delta = c_{\mathbf{B}}^T \bar{A} - c = c_{\mathbf{B}}^T B^{-1}A - c$, ami a bázis koordinátaiban nulla.

2.1.8. Definíció (Optimális bázis). A \mathbf{B} bázis optimális, ha megengedett és teljesül, hogy a redukált költség nemnegatív, vagyis hogy a $c_{\mathbf{B}}^T B^{-1}A \geq c$, azaz a szimplex módszer leáll.

2.1.9. Tétel. Ha a redukált költség nemnegatív, akkor a megoldásunk optimális.

Tehát azt tudjuk, hogy ha van egy optimális bázisunk, akkor a hozzá tartozó bázismegoldásunk optimális lesz. De ha van egy optimális bázismegoldásunk, akkor még nem biztos, hogy a bázisunk is optimális. Ez történik például a degenerált feladatnál, amikor a szimplex ciklizál. Ekkor hasznos nagyon a duál-megengedett bázis fogalma és a hozzá kapcsolódó tétel, ehhez azonban előbb érdemes definiálni az árnyékár vektort.

2.1.10. Definíció (B bázishoz tartozó árnyékár vektor). Az $y_{\mathbf{B}} = c_{\mathbf{B}}^T B^{-1}$ vektort a \mathbf{B} bázishoz tartozó árnyékár vektornak nevezzük.

2.1.11. Definíció (Duál-megengedett bázis). Egy \mathbf{B} bázist duál-megengedettnek nevezünk, ha a hozzá tartozó $y_{\mathbf{B}}$ árnyékár vektorra teljesül, hogy $y_{\mathbf{B}}^T A \geq c$.

2.1.12. Állítás. *Ha egy bázis megengedett és duál-megengedett, akkor optimális.*

A lineáris programozási feladatoknál nagyon hasznos, ha létezik olyan egyszerűen ellenőrizhető módszer, aminek segítségével könnyen meggyőződhetünk egy x megengedett megoldás optimalitásáról, valamint ha nem optimális, akkor szükségünk van egy olyan eljárásra, amellyel találhatunk egy nála jobbat. Erre alkalmas a primál-duál feladat, és a dualitás tételek.

Az eredeti LP feladatot primál feladatnak nevezzük. Minden ilyennek felírható a párja, a duál feladat.

Primál feladat (P)	Duál feladat (D)
$Ax = b$	$y^T A \geq c$
$x \geq 0$	$y \geq 0$
$\max c^T x$	$\min y^T b$

2.1.13. Megjegyzés. *A fent említett árnyékár vektor nem más, mint a \mathbf{B} bázishoz tartozó duál megoldás.*

Azt tehát már tudjuk, hogy mi a kapcsolat a primál és a hozzá tartozó duál feladat között. De mi lesz a kapcsolat a megoldásaik között?

2.1.14. Tétel (Dualitás tétel, speciális alak). *Legyen x megengedett megoldása a (P) primál feladatnak, y pedig legyen a (D) duál feladat egy megengedett megoldása. Ekkor mindkettőnek létezik optimális megoldása, és a primál optimalizálási feladat maximuma megegyezik a duál optimalizálási feladat minimumával.*

Az eddig leírtak nagyon hasznosak lesznek a többtermékes folyamokhoz kapcsolódó oszlogenerálós algoritmushoz és a dekompozíciós módszerhez is, melyeket a következő alfejezetekben ismertetek.

2.2. Többtermékes folyamok

Legyen adott egy $G = (V, E)$ irányított gráf és k termék. Minden $1 \leq i \leq k$ termékhez tartozik egy s_i forrás, ahonnan küldjük, egy t_i cél, ahova el szeretnénk juttatni, és a termékből igényelt d_i mennyiség. Feladatunk, hogy minden i termékhez megtaláljuk azt a d_i nagyságú x^i folyamot s_i -ből t_i -be, amire teljesül, hogy $\sum_{i=1}^k x_e^i \leq c_e$ minden $e \in E$ élre, ahol a c_e nem más, mint az irányítatlan gráf élein adott nemnegatív kapacitásfüggvény. Az (s_i, t_i) pontpárok természetesen különbözőek.

A többtermékes folyamok között megkülönböztetünk egész és tört értékű folyamokat. Az egész értékű változatban a folyam minden élen csak egész értékeket vehet fel, azaz $x_e^i \in \mathbb{Z}$ minden $0 \leq i \leq k$ és $e \in E$ esetén. A tört értékű folyamoknál ilyen megkötés nincs.

A továbbiakban a súlyozatlan törtváltozat megoldásáról lesz szó.

2.2.1. A Ford-Fulkerson oszlogenerálós algoritmus (1958)

Ma már sokféle közelítő és heurisztikus algoritmus létezik a többtermékes folyam probléma megoldására. A súlyozatlan törtváltozat megoldásához a Ford-Fulkerson-féle oszlogenerálós algoritmust alkalmazzuk.

Jelölje $\delta_{x^i}(v)$ a v csúcsba belépő éleken, $\varrho_{x^i}(v)$ pedig a v csúcsból kilépő éleken az x_i értékek összegét, és írjuk fel a folyamra vonatkozó feltételeket:

$$\delta_{x^i}(v) - \varrho_{x^i}(v) = 0 \quad \forall i = 1, 2, \dots, k, v \in V - \{s_i, t_i\} \quad (2.1)$$

$$\delta_{x^i}(s_i) - \varrho_{x^i}(s_i) = -d_i \quad \forall i = 1, 2, \dots, k$$

$$\delta_{x^i}(t_i) - \varrho_{x^i}(t_i) = d_i \quad \forall i = 1, 2, \dots, k$$

$$\sum_{e=1}^k x_e^i \leq c_e \quad \forall e \in E$$

$$x_e^i \geq 0 \quad \forall i = 1, 2, \dots, k, e \in E$$

A súlyozatlan tört értékű folyam probléma tulajdonképpen ekvivalens azzal, hogy a fenti egyenlőtlenségrendszernek van-e megoldása. Tehát nincs más dolgunk, mint megoldani ezt a lineáris programot. A feladat azonban nem könnyű, hiszen ha k egy nagy szám, akkor viszonylag kicsi gráfok esetén is rengeteg sora lesz ennek az LP-feladatnak, ami így túl nagy méretű ahhoz, hogy belátható időn belül a megoldás végére érjünk. Ezért hívjuk segítségül Ford és Fulkerson oszlopgenerálás algoritmusát, ami egy az eredetivel ekvivalens, de kevesebb sorból és több oszlopból álló lineáris programot ír fel. Szerencsére ez nem ront a helyzetünkön, épp ellenkezőleg, ugyanis a módszer lényege és legnagyobb előnye, hogy a változóknak mindig csak a részével kell dolgoznunk, amelyre éppen szükségünk van.

Ha létezik d_i nagyságú, x^i folyam s_i -ből t_i -be, akkor létezik $y^i \leq x^i$ folyam is, amely előállítható $s_i \rightarrow t_i$ utak nemnegatív, lineáris kombinációjaként. Feltehető, hogy minden x^i folyam ilyen. Legyen az $s_i \rightarrow t_i$ utak száma n_i , valamint jelölje z_{ij} az i termékből a j -edik úton át küldött mennyiséget, és P^{ij} az i termékhez tartozó j -edik $s_i \rightarrow t_i$ utat. Alkalmazzuk még a $\chi_e^{P^{ij}}$ jelölést, ahol χ^P a P út élekre vonatkozó karakterisztikus vektora, melynek hossza az élek száma. A vektorban egy él helyén 1-es szerepel, ha az benne van a P útban, és 0, ha nincs benne. Ezen jelölések segítségével felírhatunk egy (2.1) -gyel ekvivalens feladatot:

$$\sum_{i=1}^k \sum_{j=1}^{n_i} z_{ij} \chi_e^{P^{ij}} \leq c_e \quad \forall e \in E \quad (2.2)$$

$$\sum_{j=1}^{n_i} z_{ij} = d_i \quad i = 1, 2, \dots, k$$

$$z_{ij} \geq 0 \quad \forall i, j$$

Míg a (2.1)-ben a sorok, addig itt a változók, azaz az oszlopok nagy száma nehezítené a hagyományos úton történő megoldást.

Adjunk az eredeti gráfunkhoz k darab új csúcsot: s'_1, s'_2, \dots, s'_k , és ezeket kössük össze az eredeti csúcsokkal a következőképpen: $s'_1s_1, s'_2s_2, \dots, s'_ks_k$. Így nemcsak új csúcsokat, de k darab új élt is nyertünk. Rendeljük hozzá ezekhez a d_1, d_2, \dots, d_k , kapacitásokat, majd alkalmazzuk erre az új gráfra az előbbi (2.2)-es egyenlőtlenség-rendszert, és alakítsuk át maximalizálási feladattá, azaz egészítsük ki azzal, hogy $\sum_{i=1}^k \sum_{j=1}^{n_i} z_{ij}$ maximális értékét keressük. Ha ez a maximum egyenlő a $\sum_{i=1}^k d_i$, akkor az eredeti feladatnak is van megoldása, különben nincs.

Ennek a kibővített feladatnak a felírása ekvivalens a következővel:

$$\sum_{P \in H} x_P \chi_e^P \leq c_e \quad \forall e \in E \quad (2.3)$$

$$x_P \geq 0 \quad \forall P \in H \quad (2.4)$$

$$\max \sum_{P \in H} x_P$$

ahol H az összes (s'_i, t_i) út halmaza.

Adjunk minden kapacitás megszorítást jelentő sorhoz egy-egy eltérésváltozót. Az így kapott egyenlőtlenség-rendszert mátrixos alakban megadva, a következő felíráshoz jutunk:

$$[AI]x = c \quad (2.5)$$

$$x \geq 0$$

$$\max w^T x$$

ahol A az él-út illeszkedési mátrix. A sorai lineárisan függetlenek, mert az utólag hozzáadott k darab él egyike sem szerepelhet olyan útban, amelyikben egy másik ilyen már benne van. Így A egy teljes sorrangú mátrix lesz. A $w \in \mathbb{R}^{E \times H}$ vektor pedig olyan, aminek minden úthoz tartozó komponense 1, az élekhez tartozó komponensei pedig nullák.

Tegyük fel, hogy léteznek olyan $H' \subseteq H$ és $E' \subseteq E$ részhalmazok, melyekre $|H'| + |E'| = |E|$, és a hozzájuk tartozó $[AI]$ -beli, lineárisan független oszlopok indexeinek halmaza \mathbf{B} . Ekkor ezek az oszlopok egy bázist alkotnak, és a \mathbf{B} bázishoz tartozó bázismegoldás az az $x_{\mathbf{B}}$ vektor lesz, amely a nem \mathbf{B} -beli indexekben 0, különben pedig egyértelmű megoldása a $Bx_{\mathbf{B}} = c_{\mathbf{B}}$ egyenletnek, ahol B a \mathbf{B} bázishoz tartozó bázismátrix.

Ezen a ponton van szükségünk a szimplex módszerre a probléma megoldásához. Ám ahogy korábban említettem, a feladat túl nagy mérete miatt ez rengeteg időbe kerülne, így megmutatom, hogyan lehet ezt csupán a szükséges oszlopok felírásával orvosolni.

(2.5)-nek a $B = I$ egy azonnal adódó nemnegatív primál-megengedett bázisa. Azt, hogy egy bázismegoldás optimális-e vagy sem, a következőképpen ellenőrizhetjük: Először oldjuk meg a \mathbf{B} -hez tartozó program $y_{\mathbf{B}}^T B \geq w_{\mathbf{B}}$ duálisát, ahol $w_{\mathbf{B}}$ a w vektor \mathbf{B} bázishoz tartozó része. Az így kapott $y_{\mathbf{B}}$ vektor nemcsak a duál feladathoz tartozó megoldás, hanem az előző fejezetben említettek alapján a \mathbf{B} bázishoz

tartozó árnyékár vektor is, tehát $y_{\mathbf{B}} = w_{\mathbf{B}}^T B^{-1}$. Ekkor a primál feladat $B = I$ kezdő bázisához tartozó redukált költség a következőképpen írható fel:

$$\delta_j = w_{\mathbf{B}}^T B^{-1} A_j - w_j = y_{\mathbf{B}}^T A_j - w_j \quad j = 1, 2, \dots, |H|$$

Ellenőrizzük, hogy ez a redukált költség minden bázison kívüli j oszlopra nemnegatív-e. Ha igen, akkor a \mathbf{B} bázisunk a definíció szerint duál-megengedett, tehát optimális is, és a hozzá tartozó $x_{\mathbf{B}}$ megoldás is optimális lesz. Ha pedig van olyan j , amire a redukált költség negatív, akkor átlépünk egy másik bázisra, és iteráljuk az eljárást egészen addig, amíg optimális bázisra nem jutunk.

A redukált költségek vizsgálatánál alapvetően kétféle feltételt kell figyelembe venni.

Az egyik a \mathbf{B} -ben benne nem levő élekhez tartozó, I -beli oszlopokból származó feltétel, mely egyszerűen ezen y változó nemnegativitását mondja ki. Ezt könnyű ellenőrizni. Ha találunk negatív értékű változót, akkor egyszerűen azt mondjuk, hogy legyen ez a bázisba bemenő oszlop indexe.

A másik feltétel a B -ben benne nem lévő utakhoz kapcsolódik. Itt is azt kell ellenőrizni, hogy a redukált költség egy adott P útra nemnegatív-e, vagyis hogy az $y\chi^P$ skaláris szorzat 1-nél nagyobb egyenlő-e, hiszen az ilyen oszlopok alatt w értéke mindenhol 1. Ezek száma azonban a gráf méretéhez képest akár exponenciálisan nagy is lehet, így egyesével nehéz lenne ezt ellenőrizni. Ellenben egyszerre mindez könnyen elvégezhető: ki kell számítani minden i termékre a legrövidebb (s_i, t_i) utat például a Dijkstra-algoritmussal, y költségfüggvényre nézve. Ha mindegyikre teljesül, hogy $y\chi^P \geq 1$, akkor az összes többi, náluk hosszabb útra is teljesül ez a feltétel. Ha pedig nem mind ilyen, akkor találtunk egy olyan P utat, amire a duális feltétel megsérül, és ekkor az ehhez tartozó oszlopot visszük be a bázisba. Fontos, hogy ilyenkorra már megvizsgáltuk az élsúlyok nemnegativitását, azaz minden y -ről tudjuk, hogy nagyobb vagy egyenlő, mint nulla, tehát alkalmazhatjuk a Dijkstra-algoritmust a legrövidebb utak kereséséhez.

Most már tudjuk, hogy mi alapján visszük be az oszlopokat a bázisba. De hogyan határozzuk meg azt az indexet, amelyhez tartozó oszlopnak el kell hagynia a bázist?

Ha a bejövő index az e élhez tartozik, akkor a $Bx = 1_e$ egyenletet kell megoldanunk, ha pedig az index a P úthoz tartozik, akkor a $Bx = \chi^P$ egyenletet. A megfelelő egyenlet megoldása után a szimplex módszernél tanult hányados-szabály segítségével kiválasztjuk annak a bázisban lévő oszlopnak az indexét, aminek a helyére az előbbi módszerekkel meghatározott oszlopot vesszük be. Az eljárást addig ismételjük, míg optimális megoldáshoz nem jutunk. Ez az algoritmus, akár a hagyományos szimplex módszer, szintén ciklizálhat, de hála a lexikografikus választási szabálynak, ez itt is elkerülhető.

Ez volt tehát a Ford-Fulkerson oszlopgenerálás algoritmus, mely a nevét onnan kapta, hogy nem tekintjük az összes oszlopot, csupán azokat, amelyekre ténylegesen szükségünk van. Nagyon szorosan kapcsolódik a következő alfejezetben tárgyalt a Dantzig-Wolfe dekompozícióhoz.

2.3. A Dantzig-Wolfe dekompozíció

A dekompozíciós eljárások nagyon hasznosak nagyméretű lineáris programozási problémák megoldásához. Ezek az eredeti feladat megoldását több kisebb, vagy egyszerűen csak könnyebben kezelhető részfeladat általában többszöri megoldására vezetik vissza. Ez azért is praktikus, mert a részeredményekből kiindulva

sokkal egyszerűbben végezhető további számítások. Segítségükkel nemcsak gyorsabban juthatunk hozzá a keresett eredményekhez, de a hibákat is könnyebb megtalálni a modellekben. Gyakran alkalmazzák őket szervezési és irányítási problémákhoz, például nagyvállalatok több időszakot felölelő termelésének ütemezésénél, de egyéb közgazdasági témáknál is szívesen felhasználják őket.

A dekompozíciós módszerek között az első a Dantzig-Wolfe dekompozíció volt [4]. 1960-ban publikálták, és a megalkotói nevét viseli. Alapgondolata, hogy egy lineáris programozási feladat megoldása egyenlő a poliédere kanonikus felbontásából adódó extrémális feladat megoldásával.

Tekintsük a következő alakú lineáris programozási feladatot:

$$\begin{bmatrix} A_1 & A_2 & A_3 & \dots & A_r \\ D_1 & 0 & 0 & \dots & 0 \\ 0 & D_2 & 0 & \dots & 0 \\ 0 & 0 & D_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & D_r \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_r \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_r \end{bmatrix}$$

$$x_j \geq 0 \quad j = 1, 2, \dots, r \quad (2.6)$$

$$\max z = \sum_{j=1}^r c_j^T x_j$$

ahol A_j $m_0 \times n_j$, D_j pedig $m_j \times n_j$ méretű mátrixok, x_j és c_j olyan vektorok, amik n_j komponensből állnak, b_0 vektornak m_0 , b_j -nek pedig m_j eleme van, ahol $j = 1, 2, \dots, r$.

Ha $A_j = 0$ minden j -re, akkor a feladat sokkal egyszerűbb, mivel csak r darab független lineáris programozási feladattal állunk szemben: $D_j x_j = b_j$, ahol $x_j \geq 0$, és a célfüggvény: $\max z = c_j^T x_j$, ($j = 1, 2, \dots, r$). Ha viszont nem minden $A_j = 0$, akkor már sokkal bonyolultabb a helyzet, hiszen a $\sum_{j=1}^r A_j x_j = b_0$ összekötő feltételek (linking constraints) minden változót tartalmaznak. A (2.6)-os feladat megoldásához így egy $\sum_{j=0}^r m_j$ méretű bázismátrixszal kellene dolgozni a szimplex módszer során. Emiatt hívjuk segítségül az említett dekompozíciós eljárást.

Tegyük fel, hogy a $D_j x_j = b_j$, $x_j \geq 0$ feladat megoldásai korlátos poliédert alkotnak. Egy korlátos poliéder minden eleme megadható az extrémális pontjainak nemnegatív, lineáris kombinációjaként. Jelölje a $D_j x_j = b_j$, $x_j \geq 0$ által definiált korlátos poliéder extrémális pontjait p_{jk} , ahol $k = 1, 2, \dots, l_j$. Ekkor a $D_j x_j = b_j$ feladat bármely x_j megengedett megoldása megadható az alábbi alakban:

$$x_j = \sum_{k=1}^{l_j} \lambda_{jk} p_{jk} \quad (2.7)$$

ahol

$$\sum_{k=1}^{l_j} \lambda_{jk} = 1$$

$$\lambda_{jk} \geq 0, \quad k = 1, 2, \dots, l_j$$

Ha egy ilyen alakú x_j vektorral találkozunk, az megengedett megoldása lesz az előbbi $D_j x_j = b_j$, $x_j \geq 0$ feladatnak, hiszen a korlátos poliéder extrémális pontjainak minden nemnegatív lineáris kombinációja eleme a poliédernek.

Tekintsük a $\lambda_{jk} \geq 0$ változók olyan halmazát, mely kielégíti az alábbi feltételeket, amiket (2.7)-ből és a $\sum_{j=1}^r A_j x_j = b_0$ egyenletből kapunk:

$$\sum_{j=1}^r \sum_{k=1}^{l_j} (A_j p_{jk}) \lambda_{jk} = b_0 \quad (2.8)$$

$$\sum_{k=1}^{l_j} \lambda_{jk} = 1 \quad j = 1, 2, \dots, r$$

Ekkor a (2.8) megegyezik a (2.6)-os lineáris programozási feladat feltételeivel, hiszen utóbbi minden megengedett megoldása meghatároz egy $\lambda_{jk} \geq 0$ változó halmazt, ami eleget tesz a (2.8)-nak, és visszafelé is teljesül, hogy a (2.8)-at kielégítő $\lambda_{jk} \geq 0$ halmaz megad egy olyan $x_j \geq 0$ vektor halmazt, amely megoldása lesz a kiindulási feladatnak. Egyértelműségről nem beszélhetünk mindkét irány esetén, ugyanis a (2.6) egy megengedett megoldása nem feltétlenül egyértelműen határozza meg a (2.8)-at kielégítő $\lambda_{jk} \geq 0$ halmazt, de abban biztosak lehetünk, hogy létezik ilyen. Azért, hogy a (2.6) teljesen ekvivalens legyen a (2.8)-cal, utóbbit kiegészítjük a

$$\max z = \sum_{j=1}^r \sum_{k=1}^{l_j} (c_j^T p_{jk}) \lambda_{jk} \quad (2.9)$$

célfüggvénnyel. Így kapjuk az úgy nevezett főfeladatot (master problem):

$$\sum_{j=1}^r \sum_{k=1}^{l_j} A_j p_{jk} \lambda_{jk} = b_0 \quad (2.10)$$

$$\sum_{k=1}^{l_j} \lambda_{jk} = 1 \quad j = 1, 2, \dots, r \quad (2.11)$$

$$\lambda_{jk} \geq 0 \quad j = 1, 2, \dots, r, \quad k = 1, 2, \dots, l_j$$

$$\max z = \sum_{j=1}^r \sum_{k=1}^{l_j} (c_j^T p_{jk}) \lambda_{jk}$$

Ennek a feladatnak a legnagyobb előnye, hogy sokkal kisebb bázismátrix szükséges a megoldásához, mint az eredetihez, hiszen itt csak $m_0 + r$ korlátozó feltétel van. Viszont több változót használ fel, mint a (2.6)-os lineáris programozási feladat, hiszen a $D_j x_j = b_j$ megoldásaihoz tartozó korlátos poliédernek több extrémális pontja van, mint ahány komponense egy x_j vektornak. Ezért nem is lenne szerencsés kiszámítani az összes extrémális pontot, hogy meghatározhassuk a $A_j p_{jk}$ és $c_j^T p_{jk}$ értékeket. De nem is

kell, ha észre vesszük, hogy ilyen esetekben a szimplex módszernél bizonyos együtttható oszlopok sohasem kerülnek be a bázisba, és így a megoldásba sem. Ezért elég csupán akkor kiszámolni ezeket az értékeket, amikor ténylegesen szükség van rájuk. Ez az algoritmus a már ismertetett oszlopgenerálás. Hogyan is néz ki ennek az LP-feladatnak az együtttható mátrixa és az egyenletrendszer jobb oldala?

$$\begin{bmatrix} A_1 p_{11} & \dots & A_1 p_{1l_1} & A_2 p_{21} & \dots & A_2 p_{2l_2} & \dots & A_r p_{r1} & \dots & A_r p_{rl_2} \\ 1 & \dots & 1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 1 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 1 & \dots & 1 \end{bmatrix}$$

A jobb oldal pedig: $b = [b_0, 1, 1, \dots, 1]$ alakú. Az egyszerűség kedvéért alkalmazzuk a célfüggvényünk együttthatóira a következő jelölést

$$f_{jk} = c_j^T p_{jk}$$

Tegyük fel, hogy \mathbf{B} ennek a lineáris egyenletrendszernek egy bázisa, és legyen $\lambda_{\mathbf{B}}$ a bázishoz tartozó változókból álló vektor, $f_{\mathbf{B}}$ pedig a célfüggvény azon együttthatóiból álló vektor, melyek megfelelnek a bázisindexeknek.

Tegyük fel azt is, hogy $\lambda_{\mathbf{B}}$ egy megengedett bázismegoldás. Ekkor \mathbf{B} megengedett bázis, és $\lambda_{\mathbf{B}} = B^{-1}b$, ahol B a \mathbf{B} bázishoz tartozó bázismátrix, a célfüggvényünk értéke pedig:

$$z = f_{\mathbf{B}}^T \lambda_{\mathbf{B}}$$

Ahhoz, hogy megtudjuk, hogy a \mathbf{B} bázisunk optimális-e, vizsgáljuk meg, hogy duál-megengedett-e.

Jelölje ω_i ($i = 1, 2, \dots, m_0$) a (2.10)-es korlátozó feltételhez tartozó duális változókat, μ_j ($j = 1, 2, \dots, r$) pedig a (2.11)-hez tartozó duál változó vektorokat. A duál feladat megoldásaként megkapjuk az $f_{\mathbf{B}}^T B^{-1}$ árnyékár vektort, aminek a segítségével kiszámítható a redukált költség:

$$\delta_{jk} = f_{\mathbf{B}}^T B^{-1} A_j p_{jk} - f_{jk} = \omega^T A_j p_{jk} + \mu_j - c_j^T p_{jk} = (\omega^T A_j - c_j) p_{jk} + \mu_j \quad (2.12)$$

Ha a redukált költség mindenhol nemnegatív, akkor a bázisunk duál-megengedett, és így a (2.1.12) -es állítás szerint optimális is. Ha nem, akkor további iterációkat kell végezni. Ennek eldöntéséhez számítsuk ki a $\min_{\forall j,k} \delta_{jk}$ legkisebb redukált költséget. Ezt a következőképpen tehetjük:

$$\min_{\forall j,k} \delta_{jk} = \min \left[\min_{1 \leq k \leq l_1} \delta_{1k}, \min_{1 \leq k \leq l_2} \delta_{2k}, \dots, \min_{1 \leq k \leq l_r} \delta_{rk} \right]$$

A (2.12)-es képletből jól látszik, hogy egy adott j értékre a $\min_k \delta_{jk}$ mindig a $D_j x_j = b_j$ lineáris egyenletrendszer megengedett megoldásaihoz tartozó korlátos poliéder extrémális pontjain valósul meg. És mivel a $D_j x_j = b_j$ egyenletrendszernek minden extrémális pontja egyben egy megengedett bázismegoldása is, így a $\min_k \delta_{jk}$ optimumot a μ_j és következő feladat megoldásának összegeként kapjuk meg:

$$\begin{aligned} D_j x &= b_j \\ x_j &\geq 0 \end{aligned} \quad (2.13)$$

$$\min M_j = (\omega^T A_j - c_j)x_j$$

Tehát a feladat optimális bázismegoldásaként megad egy olyan p_{jk} extrémális pontot, amelyre az adott j értékhez tartozó δ_{jk} minden $1 \leq k \leq l_j$ -re nézve a legkisebb lesz. Ahhoz, hogy minden j -re kiszámítsuk a hozzá tartozó optimumot, r darab ilyen (2.13) alakú LP-feladatot kell megoldanunk. Tehát, ha minden részfeladatban M_j^* jelöli az M_j optimumát, akkor

$$\min_{\forall jk} \delta_{jk} = \min_j (M_j^* + \mu_j)$$

adja meg a legkisebb redukált költséget. Amennyiben ez nemnegatív, akkor a megoldás optimális, hiszen a nála nagyobb redukált költségek is nemnegatívak lesznek. Ha viszont negatív, vigyünk be ezt az oszlopot a bázisba. Azt a bázisindexet, amihez tartozó oszlop helyére a most kiválasztottat bevisszük, az oszlopgenerálásnál is említett hányados-szabály alapján számíthatjuk ki. Ezt az eljárást ismételjük addig, amíg optimális megoldáshoz nem jutunk.

A szimplex módszer elmélete alapján biztosan leáll az algoritmus, és megtaláljuk az optimumot.

Jól látható tehát, hogy ezzel a dekompozíciós algoritmussal sokkal gyorsabban és egyszerűbben kiszámítható egy nagyobb méretű LP-feladat megoldása, így ismerete rendkívül jó szolgálatot tesz majd a következő fejezetben, amiben a tengeri szélfarmok karbantartásának ütemezéséről lesz szó.

3. fejezet

Optimalizálási modellek tengeri szélfarmok karbantartásának ütemezéséhez

3.1. Bevezetés

Napjainkban az energia rendkívül fontos szerepet tölt be életünkben, így az energiaforrások, és ezen belül is inkább a megújuló energiaforrások iránti kereslet egyre nő. Utóbbiak közül kiemelhető a szélenergia. Szélenergiaforrásokkal évtizedekig csupán a szárazföldjeinken találkozhattunk, a tengeri, vagy másként offshore szélfarmok fejlesztése az utóbbi húsz évben vált szignifikánssá. Ennek az egyik legfőbb oka, hogy itt a turbinák több elektromosságot képesek előállítani, hiszen az átlagos szélesség a tengeren sokkal nagyobb, mint a parton vagy attól beljebb. Ezzel együtt azonban az ilyen offshore szélenergiaforrások üzembe helyezése, fenntartása és karbantartása is jóval többbe kerül, hiszen a tengeren több erőforrás és fejlettebb infrastruktúra szükséges mindezekhez. Egy 2009-es megállapítás szerint a fenntartási és karbantartási költségek a kiadásoknak akár a negyedét is kitehetik, így ezeket mondhatjuk a legdrágább komponenseknek. Ráadásul a természet adta körülmények miatt bizonyos munkálatok csak szűk időkorláton belül végezhetők el, melynek figyelmen kívül hagyása akár az energiatermelés megszakítását is eredményezheti, és ezáltal jelentős veszteséget is okozhat. A költségek redukálásának és a nagyobb hatékonyság elérésének érdekében a karbantartási munkálatok és az ehhez alkalmazott hajók útvonalának optimalizálása nagyon fontos. Mindez a szélenergiaforrások méretének növekedésével és a parttól való távolodásával egyre nagyobb szerepet kap, és egy egyre komplexebb feladat mielőbbi megoldását követeli a szakemberektől.

Az European Committee for Standardization szerint a karbantartási tevékenységeket a szélenergiaforrások esetében 3 nagy csoportra bonthatjuk: előre tervezett, megelőző karbantartásra, a turbinák állapota alapján történő karbantartásra és a javító jellegű, helyre állító karbantartásra.

A turbinák vizsgálatának esedékességét gyakran még a beüzemeléskor meghatározzák. Ezek általában bizonyos időszakonként, vagy például a szélkerekek korától függően kerülnek sorra. Az ilyen előre tervezett

karbantartás lényege a teljesítménycsökkenés megelőzése és a meghibásodások kockázatának csökkentése. A második típusú karbantartás a turbinák állapotának ellenőrzésén, és hosszas megfigyelésükön alapszik. A javításra akkor kerül sor, ha az indikátorok a kondíciójuk romlását és az esetleges hibák valószínűségének növekedését jelzik. A harmadik esetben a szakemberek felméri a szélerőmű állapotát, és az esetlegesen detektált hibákat kijavítják a normál üzemi körülmények helyre állítása érdekében.

Típusonként több jelentős munka is készült már a szárazföldi szélfarmok szervizelésének optimalizálásához. A megelőző jellegű karbantartáshoz 2011-ben egy magyar csapat, Kovács, Erdős, Monostori és Viharos [7] készített egy matematikai modellt a turbinák működési jellemzőit és a rendelkezésre álló energiaforrásokat figyelembe véve, melynek segítségével meghatározható a feladat elvégzésére legalkalmasabb időpont. 2012-ben Parikh [8] alkotott egy olyan matematikai modellt, mely a megelőző karbantartás ütemezésével és a kritikus komponensek helyettesítésével ad optimális megoldást a költségek minimalizálására. Szintén ebben az évben Zhang és Chowdhury [9] nyomán született egy olyan munka, amely a teljes leállás alatt keletkező energiaveszteséget optimalizálja, és ehhez az időjárást, a műszeresek csapatát, a szállítást és a megfelelő infrastruktúrát tekinti befolyásoló tényezőnek. 2013-ban pedig egy olyan modell készült Perez-Canton és Rubio-Romero [10] által, ami a rendszer megbízhatóságát képes maximalizálni. De nem csupán a megelőző jellegű karbantartási munkához találhatunk ilyen hasznos modelleket. A javító, illetve kondíció alapú szervizelés ütemezésével foglalkozott például Besnard, Patriksson, Strömberg, Wojciechowski és Bertling [11] már 2009-ben. Ők a szélelőrejelzést vették alapul az optimalizálásnál. Rajtuk kívül még rengetegen próbálkoztak és értek el sikereket ebben a témában, miközben olyan felfedezések is születtek, mint hogy a kondíció alapján történő karbantartással csökkenthető a meghibásodások előfordulásának valószínűsége, és ezáltal sokkal több szélenergiát lehet előállítani így, mint a javító jellegű szervizelésekkel. Sajnos ezek a megoldások csak részben alkalmazhatók az offshore szélfarmok karbantartásához, mivel ott további tényezők bonyolítják a problémát. A kihívásokról bővebben a következő részben lesz szó. A későbbiekben bemutatok néhány modellt is. Ezekben csak olyan szervizelések optimalizálásáról lesz szó, ahol a karbantartandó turbinák egy halmazának a javasolt javítási periódusa előre meghatározott. A szélkerekek akár a megelőző jellegű, akár az állapot alapú karbantartás elvei szerint is kiválaszthatóak.

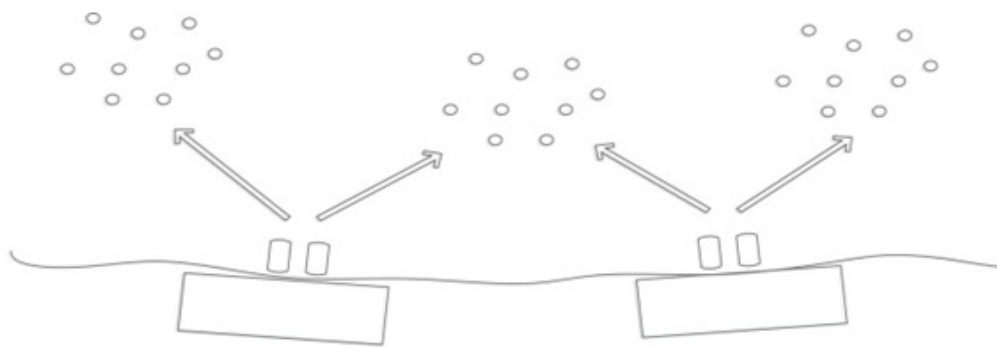
3.2. Kihívások

Az offshore szélfarmok karbantartásának ütemezése rengeteg kihívást rejt magában, hiszen számos tényezőt kell figyelembe venni. Először is szükségünk van egy olyan fogalomra, amellyel a szélerőművek elérhetőségét jellemezzük. Legyen ez a hozzáférhetőség. Ez alatt azokat az időszakokat értjük, amikor a karbantartási munkálatok nyugodtan folyhatnak. A hozzáférhetőség két meghatározó tényezője a karbantartó flotta összetétele és a környezeti viszonyok. Egy ilyen járműcsoport általában többféle hajóból, esetleg repülőkből, helikopterekből áll. Ez azért fontos és praktikus, mert a hullámok, az erős szél vagy akár a fagy különböző korlátokat szabhatnak az egyes szállítóeszközöknek. Így előfordulhat az is, hogy a rossz időjárási körülmények miatt bizonyos turbinákat akár hónapokig nem lehet megközelíteni. A járműveket azonban nem csak strapabíró képességük alapján tudjuk megkülönböztetni. Kapacitásuk mind a szállítható személyek számát, mind a felszerelések és alkatrészek összsúlyát tekintve limitált. Emiatt

gyakori probléma, hogy nagyobb létszám illetve teher esetén egy specifikusabb hajó bérlése szükséges, ami azon túl, hogy többbe kerül, a kereslet miatt csak hosszú várakozás után és nagyon rövid ideig vehető be, ezzel még tovább csökkentve az egyébként is szűk időkeretet. Bárkinek joggal fordulhat meg a kérdés a fejében: vajon miért nem bővítik a bázis gépparkját saját járművekkel. Sajnos a szélenergia termeléséből származó profit egyelőre nem elég nagy ahhoz, hogy ilyen, vagy hasonló mértékű beruházásokba kezdhessenek. Természetesen az se jó, ha a megfelelő eszközök híján meg kell szakítani a munkát, hiszen ez az energiatermelés hosszabb távú leállításával és jelentős veszteséggel járhat. A kiadásokat és a késés lehetőségét az is tovább növelheti, ha a szakemberek nem tudják befejezni a munkát egy nap alatt. Ekkor a munkaidő végén vissza kell őket szállítani a bázisra, majd a következő napon újra ki a tengerre a turbinához. Ez a probléma egyre komplikáltabb lesz, ahogy a szélfarmok távolodnak a parttól. Megoldása ismét a flotta bővítése vagy egy tengeri bázis kialakítása lenne, azonban erre a fent említett okok miatt még nincs lehetőség. Gyakori még, hogy a javítandó turbinák között fontossági sorrendet kell felállítani a karbantartás szükségessége alapján. Azzal, hogy bizonyos szélkerekek sürgősebb beavatkozást igényelnek, más turbináknál további állapotromlást, a munkálatok esetleges csúszását, és ezáltal a kitermelés további szünetelését okozhatják, amely újabb költségeket és veszteségeket jelenthet. Tehát anyagi szempontból elengedhetetlen a gondos tervezés és a határidők betartása.

A fent ismertetett nehézségek miatt fontos különösképpen, hogy megoldást találjanak a problémára. Ennek lehetőségét pedig a szakemberek a hajók útvonalának és ütemezésének optimalizálásában látják.

3.3. A feladat és megoldásai



3.1. ábra.

A 3.1-es ábra a szélfarmok elhelyezkedését szemlélteti. Minden farm turbinák egy halmazából áll, melyeket meg kell javítani a kijelölt időszak alatt. Nem messze tőlük, általában a hozzájuk legközelebb lévő kikötőben, van egy bázis, ahol a karbantartáshoz szükséges eszközök és alkatrészek beszerezhetők, és ahol a különböző hajók és a szakemberek állomásoznak. Minden szállítóeszköz más-más kapacitással rendelkezik a szállítható emberek számát és a felszerelések összsúlyát tekintve. Feltesszük, hogy elegendő jármű áll rendelkezésre a feladat elvégzéséhez, és így a szerelők magukkal tudják vinni a szükséges eszközöket és pótalkatrészeket. Adott a bázison tartózkodó hajók száma, hogy melyik milyen eszközök szállítására képes, és az időkorlát, melyet a környezeti körülmények szabnak a járműveknek (minden napra egyetlen

ilyen időszak tartozik egy járműhöz). Szükség van még olyan adatokra is, hogy a szállítás mennyi időt vesz igénybe, és hogy ez milyen költségekkel jár, valamint hogy a hajónak jelen kell-e lennie a szervizelés ideje alatt. Ha nem, akkor csupán odaszállítja a szakembereket a turbinákhoz, majd a műszak végén felveszi és visszaviszi őket a bázisra. Van információ arról is, hogy mennyi időt vesz igénybe amíg a szerelők eljutnak a hajótól a turbináig, és hogy mennyi az ő munkabérük. A karbantartással kapcsolatban könnyítésként feltesszük, hogy a javítás során új hibákat már nem detektálnak a szakemberek, azaz a folyamat közben kiderülő problémák nem befolyásolják az éppen zajló karbantartást, és így minden a tervek szerint folyhat. Az új feladatok később kerülnek beütemezésre, például ha előbb fejeződik be a tervezett munka, akkor a fennmaradó idő alkalmas lehet erre.

Nemcsak a flotta tulajdonságait kell figyelembe venni a tervezéskor, a turbináknak is vannak fontos jellemzői: ismert a karbantartás elvégzéséhez szükséges idő, a szakemberek száma, az alkatrészek és eszközök súlya, valamint hogy ezek hogyan állnak a szerelők rendelkezésére a bázison. Ezen kívül adott a tervezett időszak utolsó napja, a határidő, ameddig el kell készülnie a munkának. Ez azért fontos, mert ha a karbantartás nem fejeződik be időben, akkor büntetést kell fizetni, mely egyfajta kompenzáció a csúszásból adódó veszteségek miatt, illetve motiváció a műszerészeknek, hogy a karbantartás az ütemezésnek megfelelően történjen. Az energia ára természetesen kulcsfaktor ebben a kérdésben. A késedelmi díj kiszabható a szélfarm tulajdonosának, üzemeltetőjének vagy akár a szolgáltató bázis preferenciái alapján is. Minden turbinánál egyedi összeget jelent, és reflektál a prioritásra is, hogy melyik turbina karbantartása a legsürgősebb. Ezt a működési hibák és a biztonságosság alapján határozzák meg.

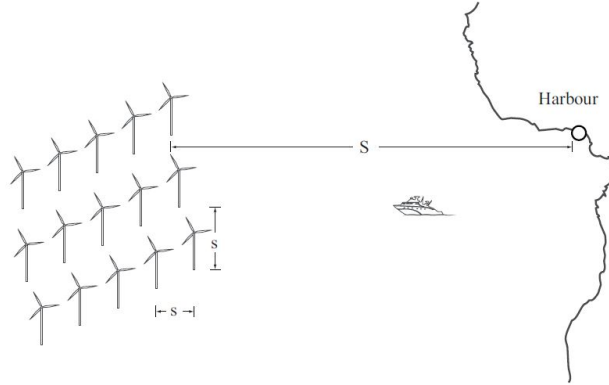
Természetesen az időkeretet is figyelembe kell venni. Ez lehet például néhány hét a nyáron, amikor nagyobb megszakítások nélkül lehet elvégezni a szervizeléseket. Fontos, hogy minden turbinát csak egyszer szervizelnek a tervezett időszakon belül.

Ezen korlátok és feltételek mellett a cél megtalálni a legalacsonyabb költségvetésű megoldást, ami egyben a termelésből adódó veszteséget is minimalizálja, természetesen a kitűzött határidők betartása mellett.

3.3.1. Első lépések egy hosszú úton

A probléma megoldására az első ötletet Lijuan Dai, Magnus Stalhane és Ingrid B. Utne [12] vetették fel 2014-ben. Modelljük minden hajóhoz hozzárendeli az optimális ütemezést és útvonalat a fent leírt korlátok figyelembe vételével. A feladatot gráfon szemléltették és az alábbi jelöléseket alkalmazták:

Legyen a karbantartandó turbinák száma n . Az i -edik turbinát az i -edik pont jelképezi, a hozzá tartozó felvételi pont pedig az $n + i$ -edik lesz. A bázist a 0 és a $2n + 1$ csúcsok jelölik. A hajók halmaza H , a tervezett időintervallum egymás után következő napok halmaza: D . Adott d napon a munkaidő maximum T_d^{max} óra lehet (pl. 12 óra). Ha letelt, a szerelők és a járművek visszatérnek a bázisra. Am ahogy említettem, a hullámok és a szél is befolyással lehetnek a hajók közlekedésére, így minden h hajóhoz tartozik egy olyan időszak az adott d napon, amikor akadály nélkül szállíthat: T_{hd}^{max} , vagyis T_d^{max} jelölés a továbbiakban T_{hd}^{max} .



3.2. ábra.

Ahogy a 3.2-es képen is látszik, a kikötő és a szélérőmű farm távolsága: S . Ez a való életben akár 100-200 km is lehet. Az egyes turbinák közötti távolság s , általában 1 km. Minden nap, a munkaidő kezdetén, a hajók S_{bi} távolságot tesznek meg a bázistól az első szélkerékig, majd innen S_{ij} távolságot a j -edik turbinához. Így a fent említett útköltség az utazással töltött idő és az időegységenkénti kiadás szorzataként számolódik. Ha a karbantartás alatt a hajó a helyszínen várakozik, akkor ez az idő is hozzáadódik az utazással töltött időhöz. Késés esetén késedelmi díjat számítanak fel a be nem fejezett szélkerekek után. Az i turbina utáni büntetés/nap: $C_i^{buntetes}$, ami magába foglalja a termelési veszteséget is, mely akár az egyes szélkerekek, akár a teljes szélfarm esetén a következő képlettel számolható:

$$C_{termveszt} = \bar{A}(t) \cdot t \cdot K_{nevleges} \cdot C_{energia}$$

ahol $\bar{A}(t)$ az az átlagos idő amíg üzemben kívül van a turbina a t periódusban, $K_{nevleges}$ a névleges kapacitás, $C_{energia}$ pedig az energia ára. Jelen esetben a cikk írói nem számoltak a hajón dolgozó emberek költségeivel.

Halmazok

- V^- : a szállítási csomópontok halmaza, $V^- = \{1, 2, \dots, n\}$
- V^+ : a felvételi pontok halmaza, $V^+ = \{n+1, n+2, \dots, 2n\}$
- $V = V^- \cup V^+$
- $V^{varakozas} \subseteq V$: azon pontok halmaza, ahol a hajóknak jelen kell lennie a karbantartás alatt
- N : az összes csomópont, beleértve a bázist, $N = V \cup \{0, 2n+1\}$
- H : a hajók halmaza (h -val indexelve)
- D : a tervezett időszak napjainak halmaza (d -vel indexelve) $D = \{1, 2, \dots\}$

Konstansok

T_{hij}	: az az idő, amíg h hajó eljut i -ből j -be,
C_{hij}	: a h hajó költsége i és j turbinák között
L_i	: az i turbina javításához szükséges alkatrészek és eszközök összsúlya
P_i	: az i turbina javításához szükséges szakemberek száma
T_i^K	: a karbantartási feladat elvégzéséhez szükséges idő az i turbinánál ($T_0^K = T_{2n+1}^K = 0$)
T_{hd}^{max}	: a h hajó munkaóráinak maximális száma a d napon
L_h^{max}	: a h hajó maximális súlykapacitása
P_h^{max}	: a h hajón szállítható személyek maximális száma
$T_i^{hatarido}$: az i turbina karbantartásának határideje
$C_i^{buntetes}$: az i turbina késedelmi díja/nap, amit a $T_i^{hatarido}$ után kell fizetni

Változók

$$x_{hijd} = \begin{cases} 1, & \text{ha } h \text{ hajó átmegy } i\text{-ből } j\text{-be a } d \text{ napon} \\ 0, & \text{különben} \end{cases}$$

y^i	: az i turbina karbantartási munkálatainak késése napokban megadva
t_{hid}	: az az időpont, amikor a h hajó odaér az i turbinához a d napon
q_{hid}	: a hajón lévő személyek száma, amikor elhagyja az i szélkereket a d napon

Matematikai modell

$$\sum_{j \in N} \sum_{h \in H} \sum_{d \in D} x_{hijd} = 1 \quad \forall i \in V \quad (3.1)$$

Ez a (3.1)-es egyenlet azt biztosítja, hogy minden turbina csak egyszer lehessen szállítási és felvételi csomópont.

$$\sum_{i \in N} x_{h0id} = 1 \quad \forall h \in H, d \in D \quad (3.2)$$

$$\sum_{i \in N} x_{hi(2n+1)d} = 1 \quad \forall h \in H, d \in D \quad (3.3)$$

A (3.2) azért felel, hogy minden hajó naponta csak egyszer hagyhassa el a bázist, a (3.3) pedig azt biztosítja, hogy minden hajó naponta csak egyszer térhessen vissza a bázisra.

$$\sum_{j \in N} x_{hjid} = \sum_{j \in N} x_{hijd} \quad \forall h \in H, d \in D, i \in N \quad (3.4)$$

A (3.4)-es összefüggés nem más, mint a folyamhoz tartozó megmaradási törvény.

$$\sum_{j \in N} x_{hijd} = \sum_{j \in N} x_{h(n+i)jd} \quad \forall h \in H, d \in D, i \in V^- \quad (3.5)$$

Ez a (3.5)-ös feltétel azt biztosítja, hogy adott turbinánál az oda-, és az elszállítás is megtörténjen még ugyanazon a napon. Vannak olyan turbinák is, ahol a hajónak meg kell várnia a helyszínen, amíg

a szervizelés befejeződik. Ekkor az adott turbinához tartozó pont egyszerre lesz szállítási és felvételi csomópont is. Erről szól a (3.6)-os egyenlet.

$$\sum_{h \in H} \sum_{d \in D} x_{hi(n+i)d} = 1 \quad \forall i \in V^{varakozas} \quad (3.6)$$

$$t_{h(n+i)d} - t_{hid} \geq T_i^K \quad \forall i \in V^-, h \in H, d \in D \quad (3.7)$$

$$\sum_{j \in N} \sum_{h \in H} \sum_{d \in D} (d \cdot x_{hijd} - y_j \leq T_i^{hatarido} \quad \forall i \in V^- \quad (3.8)$$

A (3.7) azért felel, hogy a javítással töltött idő ne lehessen kevesebb, mint amennyi a munka elvégzéséhez szükséges, (3.8) pedig azt biztosítja, hogy a karbantartás a határidőig befejeződjön.

$$(t_{hid} + T_{hij} - t_{hjd}) \cdot x_{hijd} \leq 0 \quad \forall i, j \in N, h \in H, d \in D \quad (3.9)$$

A (3.9)-es egyenlőtlenség biztosítja, hogy a hajók két pont közötti utazási ideje, valamint az érkezési időpontok reális kapcsolatban álljanak egymással, míg

$$\sum_{i \in V^-} \sum_{j \in N} L_i x_{hijd} \leq L_h^{max} \quad \forall h \in H, d \in D \quad (3.10)$$

azért felel, hogy a hajók ne vihessenek nagyobb súlyt, mint amennyi a maximális kapacitásuk.

$$(q_{hid} - P_j - q_{hjd}) \cdot x_{hijd} = 0 \quad \forall i \in V^-, j \in N, h \in H, d \in D \quad (3.11)$$

$$(q_{hid} + P_j - q_{hjd}) \cdot x_{hijd} = 0 \quad \forall i \in V^+, j \in N, h \in H, d \in D \quad (3.12)$$

(3.11) és (3.12) a szerelők számát követik nyomon a hajó adott pontba történő megérkezéskor.

$$0 \leq q_{hid} \leq P_h^{max} \quad \forall i \in N, h \in H, d \in D \quad (3.13)$$

(3.13) pedig azt biztosítja, hogy a hajón lévő létszám se negatív, se a limitnél nagyobb ne lehessen (megengedett a folyam).

Arra is ügyelni kell, hogy egy hajó se dolgozhasson többet az adott napon megengedettnél (munkaórák max. száma):

$$t_{h(2n+1)d} \leq T_{hd}^{max} \quad \forall h \in H, d \in D \quad (3.14)$$

A munkaidőt minden nap onnantól számítják, hogy a hajó elhagyta a bázist, ezt jelenti a (3.15):

$$t_{h0d} = 0 \quad \forall h \in H, d \in D \quad (3.15)$$

Fontos feltétel még a (3.16):

$$y_i \geq 0 \quad \forall i \in V^- \quad (3.16)$$

ami azt biztosítja, hogy azoknak a napoknak a száma, amivel később fejeződött be a munka, nem lehet negatív.

Végül, de nem utolsó sorban adott a bináris változó, ami azt mutatja meg, hogy a hajó aznap elment-e i -ből j -be:

$$x_{hijd} \in \{0, 1\} \quad \forall h \in H, i, j \in N, d \in D \quad (3.17)$$

A célfüggvény pedig a következő:

$$\sum_{h \in H} \sum_{i \in N} \sum_{j \in N} \sum_{d \in D} C_{hij} \cdot x_{hijd} + \sum_{i \in V^-} C_i^{buntetes} \cdot y_i \quad \forall i \in V$$

Ez volt tehát az első jelentős mérföldkő az offshore szélfarmok karbantartásának ütemezésénél. Innentől rengeteg megoldás született, bonyolultabb gondolatmenettel és számításokkal, más-más nézőpontból, de minden esetben ugyanazzal a céllal. A következő alfejezetben csupán említés szintjén kiemelnék egyet a sok közül, amely valahol félúton helyezkedik el a kezdetektől egészen napjainkig tartó, és remélhetőleg tovább folytatódó rögzös úton.

3.3.2. Innovatív módszer a megoldásban

A problémát egészen más szemszögből közelítette meg Magnus Stalhane, Lars Magnus Hvattuma és Vidar Skaar [13], mint Dai és csapata, ugyanis ők a karbantartási feladatok szempontjából vizsgálták a lehetőségeket. További különbség a két ötlet között, hogy itt az alkotók a javító és megelőző jellegű szervizeléssel foglalkoztak, a kiadásokat precízebb módon számolták ki, és hogy a modell megoldása során egy eddig még nem használt módszert, dekompozíciót alkalmaztak.

Míg az előbbi esetben a tervezett karbantartási feladatokat nem különböztették meg típusonként, addig itt külön-külön vizsgálják őket. Így felosztották őket két diszjunkt részhalmazra az alapján, hogy javító vagy megelőző jellegű az elvégzendő karbantartási munka. Ami a kettőt megkülönbözteti egymástól a mikéntjén kívül, az a termelészűnetelés költségének számítási módja. A javító jellegű karbantartás esetében a tervezett időszak kezdetétől a munka befejezéséig kalkulálódik ez az összeg, míg a megelőző jellegű karbantartásnál az az idő számít, amit a szerelők a turbinánál töltenek.

Nemcsak a munkálatokat bontották kétfelé, hanem a hajók halmazát is aszerint, hogy melyek alkalmasak az adott feladat elvégzéséhez szükséges eszközök szállítására, és melyek nem. Ezáltal viszont létrejöttek olyan részhalmazok, amik azokat a tevékenységeket foglalják magukba, amelyekhez adott hajó bevethető.

A megoldás során ők is gráfon szemléltették a problémát. Minden karbantartási feladathoz két pont tartozik. Az egyik, ahol a munka kezdődik, vagyis ahol az elvégzéséhez szükséges számú szakember elkezd a munkát, a másik, ahol a munka befejezése után fel kell venni ezeket az embereket, vagyis ahol a munkaidő végződik. A hajók útja minden esetben a bázisból indul, és ott is ér véget, így a gráf csúcsaihoz ezek is hozzátartoznak. Értelmszerűen a bázist is két pont reprezentálja. A csúcsokat élek kötik össze egymással, melyek több részhalmazra bonthatók aszerint, hogy egy hajó képes-e "átkelni" az adott élen, azaz hogy alkalmas-e azokhoz a feladatokhoz, melyeket az él összeköt. Ezekben a részhalmazokon belül is törölni kell

néhány élt, ugyanis ha egy feladat megkívánja, hogy a hajó a karbantartás ideje alatt ott várakozzon, akkor a feladathoz tartozó összes élt töröljük, kivéve azt az egyet, amely a kezdést és a befejezést reprezentáló pontokat köti össze. A részhalmazokon belül pedig minden élhez hozzárendelték az élen való áthaladás idejét és költségét.

Erre a gráfra készítettek egy él-folyam (arc-flow) modellt, aminek lényege, hogy megtalál egy minimális összköltségű, optimális utat a hálózaton keresztül. Ám az ilyen feladatok dekompozíció segítségével általában sokkal gyorsabban megoldhatóak. Így egy második út-folyam (path-flow) modellt is létrehozta, ami az előbbi Dantzig-Wolfe dekompozíció segítségével történő felbontása után adódik. Az optimális utat itt több lehetőség, előre generált útvonalból kell kiválasztani, melyekhez a költségek szintén az előzetes fázisban számolódtak. Az utak generálása heurisztikus úton történik.

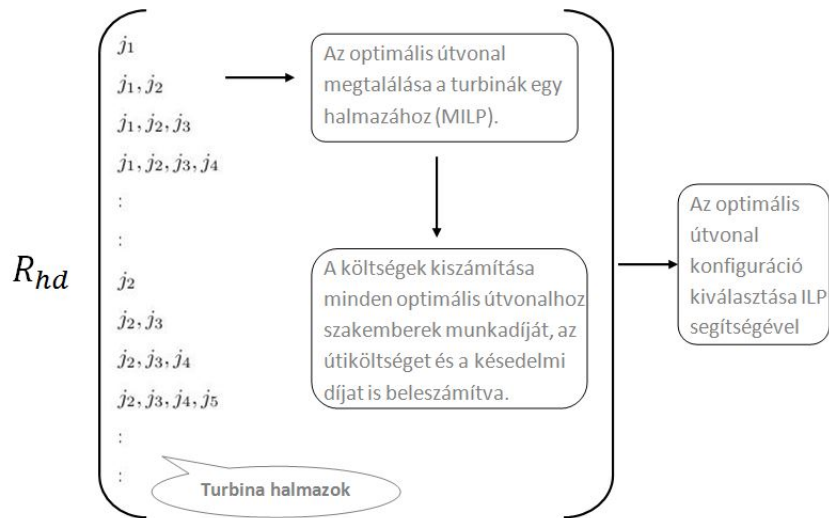
A modellek tesztelése során kiderült, hogy utóbbi az optimálishoz nagyon közeli megoldást képes adni az előbbi modellhez képest sokkal kisebb számítási nehézségek árán, kevesebb idő alatt.

Tehát ez és sok más tapasztalat is azt bizonyítja, hogy az ilyen típusú feladatok megoldásához a Dantzig-Wolfe dekompozíció egy nagyon sikeres módszer. Így az offshore szélfarmok karbantartásának ütemezéséhez és az útkeresési problémához többen is felhasználták már. A következő alfejezet is erről szól majd.

3.3.3. Egy újabb mérföldkő napjainkból

2017 januárjában jelent meg egy cikk, melyet Chandra Ade Irawan, Djamila Ouelhadj, Dylan Jones, Magnus Stalhane és Iver Bakker Sperstad [14] írtak, és amely egy olyan új megoldást mutat be a problémához, amelynél szintén felhasználták a Dantzig-Wolfe dekompozíciót. A csapat az elsőként bemutatott Dai-féle megoldás vonalát követte, ugyanakkor további feltételekkel, korlátozó tényezőkkel és a dekompozíciós módszerrel egészítették ki azt. A legfőbb különbség az eddig vizsgált megoldásokhoz képest, hogy azok készítése során a csapatok csupán egy bázist és egy szélfarmot vettek figyelembe. A gyakorlatban sajnos sokszor előfordul, hogy egyszerre több farmot kellene ellátni, így nemcsak több felszerelésre, hanem újabb bázisok létesítésére is szükség van, mely lehetővé teszi az erőforrások megosztását a szomszédos szélfarmok között, ugyanakkor még tovább bonyolítja az útkeresési és ütemezési problémát. Ám Irawan, Ouelhadj, Jones, Stalhane és Sperstad megtalálták a megoldást, melynek során nemcsak a hajók kapacitását és képességeit vették figyelembe, hanem azt is, hogy az egyes szakemberek melyik területen rendelkeznek a legnagyobb tudással. 3 kategóriába sorolhatjuk a dolgozókat a szakértelmük alapján: villamos és gépész technikus, valamint elektromechanikai műszerész. Természetesen a karbantartási munkálatok más-más szakértelmet igényelnek, ezért van ennek a kategorizálásnak is fontos szerepe.

Bár a második fejezetben már szó volt a Dantzig-Wolfe dekompozícióról, most egy ábra segítségével kicsit konkrétan is bemutatom a főbb lépéseit az útkeresési probléma megoldásánál.



3.3. ábra.

A h hajóhoz előállított összes lehetséges útvonal halmaza a d napon: R_{hd} . Első lépésként ezeket kell előállítani minden hajóhoz minden napra. Jelen esetben egy ilyen út legalább 1, legfeljebb η turbinát érinthet, ahol η jelöli a hajók által egy napon látogatható szélkerekek maximális számát. Ezután az optimális utakat kell megadni a turbinák halmazaihoz (amik legalább egy szélkereket foglalnak magukba). Ez egy vegyes, egészértékű lineáris programként (Mixed Linear Integer Program-MILP) oldható meg. Ha ez megvan, akkor ki kell számítani a hozzájuk tartozó összköltségeket, amik tartalmazzák a szerelők munkabérét, az útiköltséget és a késedelmi díjat is. Ezután ki kell választani az előbb meghatározott utak közül a legjobb út konfigurációt. Ez pedig egy egészértékű lineáris programként valósítható meg (Integer Linear Program-ILP).

A modell felírásához a következő jelölések szükségesek:

Halmazok

- B : a bázisok halmaza (b -vel indexelve)
 $H_b \subseteq H$: a b bázishoz tartozó hajók halmaza
F : szélfarmok halmaza (f -fel indexelve)
I : karbantartandó turbinák halmaza (i -vel indexelve)
 $I_f \subseteq I$: az f farmhoz tartozó turbinák halmaza
 $I_h \subseteq I$: turbinák, melyeknek a karbantartása alatt szükséges, hogy a h hajó a helyszínen várakozzon
D : napok halmaza d -vel indexelve
S : "szakmai tudás típusok" (s -sel indexelve)

Konstansok

- T_i^{hatarido} : az i turbina karbantartásának határideje
 $C_{hj'}$: a h hajó útiköltsége j és j' pontok között ($j, j' \in I \cup B$)
 $T_{hj'}$: a h hajó ennyi idő alatt teszi meg az utat j és j' pontok között ($j, j' \in I \cup B$)
 T_i^K : az i turbina karbantartásához szükséges idő
 $T_h^{\text{turbinaig}}$: ennyi idő alatt jutnak el a szerelők és a felszerelés a h hajótól egy turbináig
 L_i : az i turbina karbantartásához szükséges alkatrészek és eszközök súlya
 P_{si} : az s típusú tudással rendelkező szerelők szükséges száma az i turbinához
 P_{bsd} : a b bázison tartózkodó, s típusú tudással rendelkező szakemberek száma a d napon
 C_s : az s típusú tudással rendelkező szakemberek napidíja
 P_h^{max} : a h hajó maximális személykapacitása
 l_h^{max} : a h hajó maximális súlykapacitása
 C_i^{buntetes} : az i turbina késedelmi díja/nap, amit T_i^{hatarido} után kell fizetni
 C_h : a h hajó átlagos óránkénti üzemanyagköltsége
 T_{hdf}^{max} : a h hajó maximális munkaóráinak száma a d napon az f szélfarmon

$$\lambda_{bf} = \begin{cases} 1, & \text{ha a } b \text{ bázis látja el az } f \text{ szélfarmot} \\ 0, & \text{különben} \end{cases}$$

$$a_{hd} = \begin{cases} 1, & \text{ha a } h \text{ hajó bevethető a } d \text{ napon} \\ 0, & \text{különben} \end{cases}$$

$$g_{hi} = \begin{cases} 1, & \text{ha a } h \text{ hajó el tudja szállítani az } i \text{ turbinához szükséges alkatrészeket} \\ 0, & \text{különben} \end{cases}$$

$$m_{id} = \begin{cases} 1, & \text{ha az } i \text{ turbinához rendelkezésre áll alkatrész a } d \text{ napon} \\ 0, & \text{különben} \end{cases}$$

A lehetséges utak R_{hd} halmazához is tartozik néhány paraméter:

- L_{hdr} : turbinák rendezett listája, amelyeket a h hajó az r útvonal során a d napon érint
 C_{hdr} : a h hajó r útjához tartozó összköltség a d napon
 P_{hdrs} : a h hajón tartózkodó s tudású szakemberek száma az r út során a d napon

$$\theta_{hdr i} = \begin{cases} 1, & \text{ha a } h \text{ hajó az } r \text{ útja során a } d \text{ napon érinti az } i \text{ turbinát} \\ 0, & \text{különben} \end{cases}$$

Az ILP modell feladata meghatározni a legjobb út konfigurációt az összes lehetséges útvonal közül. A felírásához szükséges bináris változó:

$$x_{hdr} = \begin{cases} 1, & \text{ha a } h \text{ hajó } r \text{ útja a } d \text{ napon szerepel a megoldásban} \\ 0, & \text{különben} \end{cases}$$

Az ILP modell:

Egy hajó egy nap csak egy utat tehet meg, ezt biztosítja a (3.18):

$$\sum_{r \in R_{hd}} x_{hdr} \leq 1 \quad \forall h \in H, d \in D \quad (3.18)$$

$$\sum_{h \in H} \sum_{d \in D} \sum_{r \in R_{hd}} (x_{hdr} \cdot \theta_{hdr i}) = 1 \quad \forall i \in I \quad (3.19)$$

Az előbbi (3.19)-es egyenlet azt garantálja, hogy egy turbina a tervezett időszakon belül csak egyszer lehet szervizelve. A (3.20)-as feltétel azt jelenti, hogy a karbantartáshoz szükséges szakemberek típusonkénti száma nem haladhatja a bázison lévő, elérhető műszerészek számát:

$$\sum_{h \in H_b} \sum_{r \in R_{hd}} (x_{hdr} \cdot P_{hdrs}) \leq P_{bst} \quad \forall s \in S, b \in B, d \in D \quad (3.20)$$

$$x_{hdr} = \{1, 0\} \quad (3.21)$$

A célfüggvény a következő:

$$\min \sum_{d \in D} \sum_{h \in H} \sum_{r \in R_{hd}} (x_{hdr} \cdot c_{hdr}) \quad (3.22)$$

Jelentése: az optimális útvonalhoz kiválasztott utak összköltségét kell minimalizálni.

A MILP modell:

Ez a modell egy kibővítése a Dai-féle modellnek, és ez felel azért, hogy minden hajóra megtalálja az optimális utat a turbinák egy halmazához az adott napon.

Legyen n a szervizelendő turbinák száma, és legyen I' ezeknek a szélkerekeknek és az őket ellátó b bázisnak a halmaza: $I' = \{b, i_1, \dots, i_n\}$, vagy az egyszerűség kedvéért: $I' = \{0, 1, \dots, n\}$. A Dai-modell jelölései közül is felhasználunk néhányat:

V^-	: a szállítási csomópontok halmaza, $V^- = \{1, 2, \dots, n\}$
V^+	: a felvételi pontok halmaza, $V^+ = \{n+1, n+2, \dots, 2n\}$
$V^* = V^- \cup V^+ \cup \{0, 2n+1\}$: a 0 és a $2n+1$ a b bázist jelölik, mint kezdő és végpontot
$V^{varakozas} \subseteq V^-$: azon pontok halmaza, ahol a hajóknak jelen kell lennie a karbantartás alatt
T'_j	: az időpont, amikor a h hajó megérkezik a j pontba ($i \in V^*$)
p_{hsj}	: az s típusú szaktudással rendelkező szakemberek száma a h hajón, amikor az elhagyja a j pontot
q_s	: s típusú szaktudással rendelkező szakemberek száma az adott hajón

Végül egy bináris változó:

$$Y_{jj'} = \begin{cases} 1, & \text{ha a } h \text{ hajó közlekedik } j \text{ és } j' \text{ között } (j, j' \in V^*) \\ 0, & \text{különben} \end{cases}$$

Feltételek

Ahogy a Dai-féle modellben is láttuk, az első feltétel itt is arra vonatkozik, hogy minden turbina csak egyszer lehet felvételi és szállítási pont:

$$\sum_{j' \in V^*} Y_{jj'} = 1 \quad \forall j \in V^* \quad (3.23)$$

$$\sum_{j \in V^-} Y_{0j} = 1 \quad (3.24)$$

$$\sum_{j \in V^+} Y_{j(2n+1)} = 1 \quad (3.25)$$

(3.24) és (3.25) garantálják, hogy egy hajó egy nap csak egyszer hagyassa el a bázist és egyszer térhessen oda vissza.

$$\sum_{j' \in V^*} Y_{jj'} = \sum_{j' \in V^*} Y_{j'j} \quad \forall j \in V^* \quad (3.26)$$

A (3.26) a folyamhoz tartozó megmaradási feltételt írja le, míg

$$\sum_{j' \in V^*} Y_{jj'} = \sum_{j' \in V^*} Y_{(n+j)j'} \quad \forall j \in V^- \quad (3.27)$$

biztosítja, hogy minden turbinához odaszállították az embereket, és vissza is.

$$Y_{j(n+j)} = 1 \quad \forall j \in V^{varakozas} \quad (3.28)$$

(3.28) garantálja, hogy azoknál a turbináknál, ahol a hajónak jelen kell lennie a karbantartás alatt, a jármű szemléletesen közvetlenül a szállítási pontból a hozzá tartozó felvételi pontba menjen.

A következő feltétel azt biztosítja, hogy a leszállítás és a felvétel időpontja között eltelt idő nem kevesebb, mint ami a feladat elvégzéséhez szükséges.

$$T'_{(n+j)} - T'_j \geq T_j^K + T^{turbinaig} \quad \forall j \in V^- \quad (3.29)$$

$$T'_{(2n+1)} \leq T_{df}^{max} \quad (3.30)$$

A (3.30)-as korlátozás azért felel, hogy a hajók tényleges munkaóráinak száma ne lehessen nagyobb, mint a maximális munkaidő.

$$T'_0 = 0 \quad (3.31)$$

Ez a feltétel azt állítja, hogy a munkaidő onnantól számolódik, hogy a hajó elhagyta a bázist.

$$0 \leq \sum_{s \in S} p_{sj} \leq P^{max} \quad \forall j \in V^* \quad (3.32)$$

(3.32) azt garantálja, hogy a fedélzeten lévő emberek száma nem lehet több, mint a hajó személykapacitása, (3.33) pedig az utazási idő kompatibilitásáért felel.

$$Y_{jj'}(T'_j + T_{jj'} + T^{turbinaig} - T'_{j'}) \leq 0 \quad \forall j, j' \in V^*, j' \neq j + n \quad (3.33)$$

A következő két egyenlet feladata, hogy nyomon kövessék a fedélzeten lévő személyek számát az adott pontba való megérkezéskor:

$$Y_{jj'}(p_{sj} - P_{sj'-p_{sj'}}) = 0 \quad \forall s \in S, j \in V^-, j' \in V^* \quad (3.34)$$

$$Y_{jj'}(p_{sj} - P_{sj'-p_{sj'}}) = 0 \quad \forall s \in S, j \in V^+, j' \in V^* \quad (3.35)$$

$$q_s = \max_{j \in V^*} p_{sj} = p_{s0} \quad \forall s \in S \quad (3.36)$$

Az előbbi (3.36)-os egyenlet azt számítja ki, hogy adott napon hány s szaktudású szerelőnek kell a hajóra szállnia. A következő egyenlőtlenség pedig azt jelenti, hogy a karbantartáshoz szükséges szakemberek száma nem lehet nagyobb, mint a rendelkezésre álló személyek száma

$$p_{sj} \leq P_s \quad \forall s \in S, j \in V^* \quad (3.37)$$

A célfüggvény pedig:

$$\min(c^{pr} + c^{dr}) \quad (3.38)$$

ahol c^{pr} a szakemberek költsége az r út során a d napon, és c^{dr} az r úthoz tartozó útiköltség a d napon. Ezek a következőképpen számolhatók:

$$c^{pr} = \sum_{s \in S} (q_s \cdot C_s) \quad (3.39)$$

és

$$c^{dr} = \sum_{j \in V^*} \sum_{j' \in V^*} (Y_{jj'} \cdot C_{jj'}) \quad (3.40)$$

Ha sikerült meghatározni az optimális utat, akkor még ki kell számolni a hozzá tartozó költséget. Adott hajó r útjához a d napon ez az összeg legyen c^r , mely tartalmazza a büntetést is, amit c^{lr} jelöl. Kiszámításuk pedig a következő képletekkel történik:

$$c^r = c^{pr} + c^{dr} + c^{lr} \quad (3.41)$$

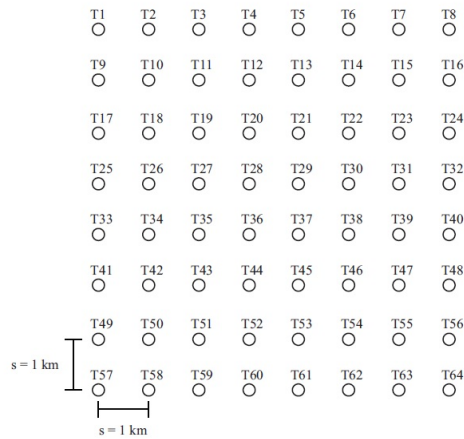
és

$$c^{lr} = \sum_i (\theta_i \cdot \max\{0, (d - T_i^{\text{hatarido}})\} \cdot C_i^{\text{buntetes}}) \quad (3.42)$$

Ezzel végére is értünk a modellek bemutatásának. Hatékonyságukat a következőekben bemutatott futási eredmények támasztják alá. Lássuk hogyan remekeltek!

3.4. Futási eredmények

Ebben a részben a modellekhez elkészített programok példákön való tesztelésének eredményeit mutatom be röviden. A Dai-féle megoldáshoz 3 adathalmazt hoztak létre: az A példában 4 turbinát kell szervizelni, a B -ben 6-ot, a C -ben pedig 8-at. Minden adathalmazon belül 10 esetet vizsgáltak, amiket random generáltak az alábbi ábrát alapul véve. Ez egy lehetséges offshore szélfarmot ábrázol, ahol a 64 turbina 8x8-as alakzatban helyezkedik el. Közöttük a távolság 1 km, a farm bázistól mért távolsága pedig 40 km.



3.4. ábra.

Mind a 10 esetben 2 hajó útját kellett megtervezni, illetve mindet tesztelték 2 és 3 napos intervallumra is, hogy a periódus megváltoztatásából adódó esetleges különbségek is láthatóak legyenek a futási időnél és a pontosságánál.

Az A példa tesztelése után a csapat megállapította, hogy minden esetben sikerült megtalálnia a programnak az optimális megoldást 1 percen belül, illetve hogy egyetlen esetben sem kellett a munka csúszása miatt büntetést fizetni. Arra is felfigyeltek, hogy 10-ből 2 esetben a tervezett időszak 2 napról 3 napra történő meghosszabbításával a célfüggvény értéke csökkenni tudott. A többenél nagyjából azonos eredményeket kaptak. Nem úgy, mint a B halmaz próbája során. Itt már 6 turbinát kellett figyelembe venni. Szerencsére minden esetben sikerült kiszámítani az optimális értéket a programnak, és 10-ből 8 esetben kisebb lett a célfüggvény minimum értéke, amikor 2 helyett 3 nappal számoltak. A maradék 2 esetben ugyanazt az eredményt kapták. Az előbbihez hasonlóan, itt sem került sor késésre, így büntetést sem kellett fizetni. Emiatt a C adathalmaz kilóg a sorból. Itt ugyanis már 8 turbinával kellett foglalkozni, ami jelentősen megnehezítette a feladatot. Ez abban is megmutatkozik, hogy nem minden esetben volt megoldható a probléma, a megoldható esetekben pedig a legtöbbször csak csúszással sikerült optimális értéket számítani. Ennek ellenére elmondható, hogy a célfüggvény értéke általában csökkent, amikor meghosszabbították a kijelölt időszakot 1 nappal, bár ez nem minden esetben jelenti azt, hogy nem kellett büntetést fizetni.

Ebből is jól látható, hogy bonyolultabb példák megoldására nem mindig képes a Dai és csapata által elkészített modell. Hogy össze tudják hasonlítani a hatékonyságot, az utolsóként bemutatott modell alkotó a Dai-féle C -vel jelölt adathalmazt alapul véve próbálta ki programját. Mivel Dai célfüggvénye csak az útiköltséget és a késedelmi díjat veszi figyelembe, így a szakemberek munkadíját itt 0-nak tekintették. Továbbá az első modell nem számol azzal, hogy hány szerelő áll rendelkezésre a bázison, ezért számukat az utóbbi modell tesztelésekor nagyon nagyra állították be, ami gyakorlatilag azt jelenti, hogy mindig van elég munkás az elvégzendő feladatokhoz. Ezen kívül variálták még az egy hajó által látogatható turbinák számát (3,4,5 és 8 turbina esetén).

Bizonyos esetekben nagyon gyorsan, máskor lassabban sikerült kiszámítani az optimális megoldást. 5 és 8 látogatható turbina esetén mind a 10 esetre megtalálta a program a minimális célfüggvény értéket. Összességében elmondható, hogy a precízebb modell segítségével sokkal könnyebben kiszámítható az optimum a C adathalmazhoz, mint a Dai-féle modellel. Más bonyolultabb példára is tesztelték a programot, hogy kihasználhassák azt a jó tulajdonságát, hogy sokkal több mindent képes figyelembe venni, mint elődje. Ehhez 2 adathalmazt ($G1$ és $G2$) generáltak véletlenszerűen, és itt is mindegyiknél 10 különböző esetet vizsgálnak. 2 bázist és 3 szélfarmot tekintettek. A bázisok itt is 2-2 különböző tulajdonságú hajóval rendelkeznek, és egyszerre legfeljebb 2 farmot képesek ellátni. Típusonként a szakemberek száma és a munkabérük is adott. Mindkét halmaznál 3 és 7 napos a tervezett karbantartási időszak, és 8 és 12 turbina igényel egyidejűleg szervizelést egy-egy szélfarmon. Több variációt is kipróbáltak: a C adathalmazhoz hasonlóan itt is 3, 4, 5 és 8 turbinát láthat el egy hajó egy nap. Az így összeállított adathalmazokkal sokkal nehezebb dolga van a programnak, hiszen jóval nagyobb, mint a C példa. Az eredmények viszont pozitívak: a legtöbb esetben a modell törekszik a büntetés elkerülésére, és így csak néhányánál kerül sor késésre. A $G1$ adathalmaz tesztje alapján állítható, hogy minél több turbinát láthat el egy nap egy hajó, annál kisebb lesz az útiköltség, a munkadíj és a büntetés is. A $G1$ adathalmaz eredményeivel szemben a $G2$ -n viszont azt tapasztalták, hogy a büntetés az összköltséghez képest csökken, ha kevesebb turbinát tud egy nap alatt meglátogatni egy hajó. Továbbá érdekességként kipróbáltak egy még nagyobb modellt, amely 3 szélfarmmal, összesen 72 szervizelendő turbinával és 2 hetes időszakkal számol. Minden hajó

legfeljebb 5 szélkerékhez tud eljutni egy nap. A tapasztalat az, hogy bár 40 óra kellett hozzá, a program sikeresen lefutott, és kiszámította az optimális célfüggvény értéket, vagyis ez a dekompozíciós modell nagyobb adathalmazon is képes jól dolgozni.

3.5. Konklúzió és további lehetőségek

Bár nagyon jól szerepelt a dekompozíciós modell, számos szempontból lehetne még javítani rajta. Szerecsére még rengeteg lehetőséget hordoz magában az offshore szélfarmok karbantartásához tartozó útkeresési és ütemezési feladat. Jó ötlet lehet például az időjárás bizonytalanságát és változékonyságát, mint döntéstámogató eszközöket figyelembe venni. Esetleg ki lehetne még bővíteni a modellt azzal, hogy a hajókhöz ne csak egy időkeret tartozzon, hanem kettő, vagy több, ezek közül valamilyen alkalmassági mérce alapján kiválasztva a szállításhoz legalkalmasabbat. Illetve a szélfarmok folyamatos fejlődése újabb kapaszkodókat adhat a fejlesztőknek, a technika további vívmányairól már nem is beszélve.

Ezen a problémán kívül több lehetőséget is rejt magában az offshore szélfarmok fejlesztése. Például kísérleteznek az alkatrészek súlyának csökkentésével, dolgoznak a problémás sebességváltók kiküszöbölésén, valamint olyan kialakítás létrehozásán, amitől a szélkerekek a tájfunokkal és hurrikánokkal szemben is ellenállóbbak lesznek. Minél több tengeri szélfarm létesül, annál több új problémával néz majd szembe ez az iparág, így a lehetőségek tárháza idővel folyamatosan bővül majd, mi pedig csak remélhetjük, hogy napjaink kutatói lépést tudnak tartani a fejlődéssel és legyőzik a kihívásokat.

Irodalomjegyzék

- [1] FÁBIÁN CSABA, KIRÁLY TAMÁS, PAPP OLGA, Jegyzet az Operációkutatás (elemző, programozó matematikus) tárgyhoz, 2015
- [2] KIRÁLY TAMÁS, KIS TAMÁS, SZEGŐ LÁSZLÓ, Online jegyzet az Egészértékű programozás I. és II. tárgyhöz, 2016. május
- [3] SZÁNTAI TAMÁS, BME Alkalmazott matematikus MSc Operációkutatás szakirány Sztochasztikus programozás, 20. előadás jegyzete, 2015
- [4] STAHL JÁNOS, Lineáris programozási dekompozíciós eljárások, Közgazdasági és Jogi Könyvkiadó, Budapest 1978
- [5] ALEXANDER SCHRIJVER, Theory of Linear and Integer Programming, 1986
- [6] GEORGE L. NEMHAUSER, LAURENCE A. WOLSEY, Integer and Combinatorial Optimization, 1999
- [7] KOVÁCS A., ERDŐS G., VIHAROS Z. J., MONOSTORI L., A system for the detailed scheduling of wind farm maintenance, CIRP Annals Manufacturing Technology, 60, 497-501, 2011
- [8] PARIKH, N. D., Optimizing maintenance cost of wind farms by scheduling preventive maintenance and replacement of critical components using mathematical approach, Texas AandM University, US., 2012
- [9] ZHANG, J., CHOWDHURY, S., Optimal preventive maintenance time windows for offshore wind farms subject to wake losses, 2012
- [10] PEREZ-CANTON, S., RUBIO-ROMERO, J. C., A model for the preventive maintenance scheduling of power plants including wind farms, Reliability Engineering and System Safety, 119, 67-75., 2013
- [11] BESNARD, F., PATRIKSSON, M., STRÖMBERG, A., WOJCIECHOWSKI, A., BERTLING, L., An optimization framework for opportunistic maintenance of offshore wind power system, In Proceedings of IEEE Bucharest powertech conference, 2009
- [12] LIJUAN DAI, MAGNUS STALHANE, INGRID B. UTNE, Routing and scheduling of maintenance fleet for offshore wind farms, 2014

- [13] MAGNUS STALHANE, LARS MAGNUS HVATTUMA, VIDAR SKAAR, Optimization of routing and scheduling of vessels to perform maintenance at offshore wind farms, 2015
- [14] CHANDRA ADE IRAWAN, DJAMILA OUELHADJ, DYLAN JONES, MAGNUS STALHANE, IVER BAKKER SPERSTAD, Optimisation of maintenance routing and scheduling for offshore wind farms, 2017
- [15] ZHENDER, ALAN, WARHAFT, ZELLMAN, "University Collaboration on Wind Energy", Cornell University, 2016