

TDK-dolgozat

Nagy Bence Bendegúz

Nagy Bence Bendegúz

ORVOSI FOLYAMATOK TÁMOGATÁSA P-GRÁF MÓDSZERTAN ALKALMAZÁSÁVAL

Témavezető: Baumgartner János

Műszaki Informatikai Kar, Rendszer- és Számítástudományi
Tanszék

Pannon Egyetem
Műszaki Informatikai Kar
2021



Tartalom

Bevezető.....	3
Bevezetés és irodalom	4
Problémadefiníció és alkalmazott módszertan	7
Problémadefiníció	7
A Módszertan Validálása.....	16
A tesztről általánosan.....	16
Első validációs teszteset – 1 vizsgálat 2 időpont.....	16
Második validációs teszteset – három vizsgálat, három – kettő – kettő időszákkal	17
Harmadik validációs teszteset – három vizsgálat, három – kettő – kettő időszákkal	20
Negyedik validációs teszteset – három vizsgálat, három – kettő – kettő időszákkal	23
Ötödik validációs teszteset – három vizsgálat, három – kettő – kettő időszákkal	25
Hatodik validációs teszteset – öt vizsgálat, négy – három – kettő – három - öt időszákkal	28
Hetedik validációs teszteset – három vizsgálat, három – kettő – kettő időszákkal	30
Irodalomjegyzék	35
Összegzés.....	36

Bevezető

Az egészségügyben nagyon sokszor előfordul, hogy egy páciens számára a kezelőorvos több vizsgálatot is előír, hiszen egy-egy adott betegség vagy probléma néha összetettebb kivizsgálást kíván meg. Ennek összhangban tartása még egy intézményen belül is nehézkes tud lenni. Abban az esetben, ha betegnek külső épületbe, netán egy független külsős helyszínen kell megjelenie, tovább bonyolítja a folyamatot. Az egyes személyek számára kért vizsgálatok ütemezése a gyakorlatban ma úgy történik, hogy egy asszisztens a rendelkezésre álló időpontok közül ad ki a beteg számára egyet-egyet. Ez több szempontból sem a legjobb megoldás. Az időpontok nem feltétlen a legjobb időkihasználtság szerint vannak létrehozva, ezen kívül emberi tényezővel is számolhatunk, hiszen előfordulhat, hogy nem megfelelő időpontra kapunk időpontot, emiatt várnunk kell. A fellépő komplikációk következményeként a mai egészségügyben sokszor találkozhatunk olyan helyzettel, hogy több órát kell várjunk egy adott rendelésen, mert a különböző hibalehetőségek összeadódva komoly ütemezési problémákat okozhatnak. Természetesen ezek közt találhatunk olyan pontokat, amelyek bármilyen rendszer használatával elkerülhetetlenek.

Bevezetés és irodalom

Az egészségügyben egyre több helyen találkozhatunk informatikai megoldásokkal, amelyek hozzájárulnak annak fejlődéséhez és modernizálásához. Megemlíthetők robotok, amelyek segítik az orvosok munkáit, esetleg az okostelefon kompatibilis diagnosztikai eszközöket. Ellenben nem szabad elmennünk a kórházak- és a betegforgalom irányítását segítő rendszereket. A széleskörben elérhető gyógyintézeti ellátás fennállása óta nagy problémát jelent a túlterheltség. Az elmúlt nagyjából 100 évben sok törekvést találhatunk, amely valamely kórházi ügymenetet próbált a lehetőségekhez mérten optimálissá tenni. Elsődleges fennakadást az egyes időpontok hosszának kiszámítása adja. Mivel minden ember és vele együtt az elváltozásai is mások így sok esetben figyelembe kell venni az egyedi kezelési igényeket, az orvosok emberi tényezőit. Előfordul, hogy helyzetben találkozhatunk olyan helyzettel, amikor egy ember állapota miatt prioritást élvez. Ezen körülmények mind nehezítik a problémára való megoldás megtalálását. Fontos megemlíteni, hogy mind a személyzetnek, mind az igénybe vett erőforrások kapacitása véges. Ide értendők a gépek, berendezések, amelyek sok esetben elengedhetetlenek a helyes diagnózis felállításához. Látható, hogy az egészségügyben rengeteg idő és pénz spórolható meg az egyes folyamatok megfelelő optimalizálásával. Az évek során más-más, már említett problémára, más és más megközelítésből próbáltak megoldást találni. Elsősorban a nem sürgősségi, járó és fekvőbeteg ellátásban történtek előre lépések, hiszen itt találkozhatunk legkisebb eséllyel terven felüli, azonnali ellátásra szoruló pácienssel.

Már az 1950-as években is tettek lépéseket annak érdekében, hogy a betegek várakozási idejét csökkentsék. A kor tudását felhasználva egymástól függetlenül Welch, Lindley és Bailey is megalkották saját módszereiket a hasonló problémára.

Nagyjából 10 évvel később az előbb említett eredmények publikálása után, 1966-ban Fetter és Thompson lineáris optimalizálási modellt segítségével dolgozott ki módszert. Ők a betegek várakozási ideje mellett már figyelmet fordítottak a dolgozók idejére is. Felismerték, hogy nagyon sok esetben úgynevezett „üresjárat” alakul ki az orvosok és ápolók munkaidejében. Figyelembe véve ezen két tényezőt, olyan modellt hoztak létre, amely mindkettőt minimalizálja.

A következő nagyobb eredményre több, mint 10 évet kellett várni, 1979-ig. Azonban, ekkor már a számítógépeket is bevonták a kutatásokba. Vissers több időpont

foglaló rendszert hasonlított össze munkája során. Tesztjének alapját az elődjeihez hasonlóan a betegek várakozási ideje adta. Úgy vélte, hogy a legjobb döntést úgy tudja hozni, hogy számítógépes szimulációt hív segítségül.

Nagyobb áttöréssel az ezredforduló környékén találkozhatunk. Podgorelec és Kokol, 1997-ben sikeresen elkészített egy olyan összetett matematikai modellt, amely alapja a genetikai algoritmus. Sokkal szélesebb körben használható, hiszen ez már figyelembe veszi a speciális terápiás igényeket is.

Megfigyelhető, hogy a technikai fejlődés mekkora hatással van életünk minden szakaszára. Erre nagyszerű példa, hogy a 2010-es évektől kezdődően indultak be leginkább az egészségügyi ellátás görbületenységet célzó tanulmányok. Mivel nagyon komplex problémákról beszélhetünk ezekben az esetekben, így a számítógépek számítási sebességének növekedésével sokkal összetettebb modellek felírása vált elérhetőbbé.

2010-ben, Zeng egy újnak számító módszert alkalmazott az előzőekben megismert problémára. Úgy vélte, hogy a túlfoglalás lehet a megoldás. Azaz több egy-egy időpontra több embert is felvesz, ezzel pedig annak a negatív hatásait próbálta csökkenteni, ha nem jelennek meg egyes páciensek. A módszer működését mi sem bizonyítja jobban, minthogy a repülőtársaságok napjainkban is alkalmazzák. Hátránya akkor érzékelhető, ha mindenki megjelenik. Ekkor kompenzációra van szükség.

2013-ban Cayirli és Gunes ismét visszatértek a '90-es években alkalmazott számítógépes szimulációhoz. Az előddel ellentétben ők a heurisztikus algoritmust választották ki. Szemléletüket a kapacitás integrálása jellemzi, amelyet próbáltak az időpontfoglalási rendszerekben hasznosítani.

2014-ben több egymástól független megoldással találkozhatunk. Ezek merőben eltérő problémát boncoltak fel. Egyrésztől megemlíthető Azadeh és Mtsai munkája, amelyet sürgősségi laborosztályok ütemezési problémáit próbálta segíteni. Itt már az online megoldások felé nyitás is megfigyelhető kezdetleges szinten. Az koncepciójuk alapját a korábban már használt genetikai algoritmus adta. Nem hagyható ki Condotta és Shakhlevich kutatási eredményei sem. Specializálódva a kemoterápiás ambuláns ellátásra, a már jól ismert célt tűzték ki. A betegek várakozási ideje mellett, a dolgozók túlterheltségi idejét minimalizálni. Az eddig megismert módszerek helyett, egy új megközelítést használtak, több szintű programozási modellt. A kor szellemének megfelelően ez is sokkal inkább számítógép közelebbi gondolkodásra utal.

A következő évben Azadeh korábbi munkáját folytatva, újra a félig online ütemező rendszerét, ám a csapat mellett a használt technológiát is megváltoztatta. Eben az évben a MILP modellre esett a választásuk. Céluk megegyezett a korábbival, ám specializálódtak a patológiai vizsgálatokra.

2016-ban, Castaing, Cohn, Denton és Weizer ismét a programozási megoldások felé nyitottak. Kétlépcsős programozási modell segítségével dolgoztak. Fő céljuk egy olyan összetett eset optimalizálása volt, ahol egy kórházi fekvőbeteg osztályon egy-egy dolgozóhoz egyszerre több beteg tartozik. Terveik közt szerepelt az egyensúly teremtése a betegek várakozási ideje és a kezelésekhez szükséges idők közt.

A 2017-2018-as évek során a korábbiakhoz képest szembetűnő mennyiséggel több megoldás született az egészségügyi folyamatok optimalizálásában. Túlnyomó többségben a már ismert problémákat vették alapul, miszerint a napi működés során a betegek várakozási idejének csökkentése mellett az orvosok tétlenségi idejét is csökkentették. Ezekkel egyetemben az erőforrások maximális kihasználása is fontos szempont a legtöbb esetben. Ezen időszakban az előzőekben ismertetett módszerek továbbfejlesztése mellett találkozhatunk új szemléletmódokkal is. Nguyen, Sivakumar és Graves heurisztikus szabályokat dolgoztak ki 2017-ben. 2018-ban ezzel szemben, Srinivas és Ravindran vényköteles elemzési keretrendszerrel vezetett be. Célja az időpontfoglalási rendszer teljesítményét. Az évtized elején felismert túlfoglalási módszer továbbfejlesztett felhasználását is tartalmazza az általuk kidolgozott eljárás. Megemlíthető a szintén 2018-ban, Lee, Heim, Sriskandarajah és Zhu által használt szekvenciális blokkütemezési eljárás, illetve a Goldstein, Hribar, Read-Brown és Chiang által használatos lineáris, vegyes-egész programozási modell.

Napjainkhoz közelítve, 2019-ben Bowen, Jiafu és Chongjun által fejlesztett programozási modellt érdemes megemlíteni, amely Benders dekompozíciójának és átlagos közelítési technikának kombinációjából alakítottak ki.

Problémadefiníció és alkalmazott módszertanok

Problémadefiníció

Ahogy az előző pontban is említettem az egészségügyi folyamatok optimalizálása már hosszú ideje egy széleskörben kutatott téma. Sok specifikus problémára olvashattunk példát. A járóbeteg ellátásban mégis a legalapvetőbb probléma a betegek várakozási ideje, ezzel párhuzamosan pedig az erőforrások kihasználtsága. Alapvetően jelenleg is a legkézenfekvőbb módszer használatos a kórházi folyamatok során. Az orvosok megbeszélés alapján adnak időpontokat, rendelésenként meghatározott időintervallumból kiválasztva. Ennek alapvető problémája a szervezetlenség. Előfordulhat, hogy több kiválasztott vizsgálat közül, teljesen véletlenszerű sorrendben kell megjelenünk rajtuk. Ebben az eljárásban nem veszik figyelembe az egyes tényezők befolyásoltságát. Gondolhatunk itt az átérési időkre, illetve a vizsgálatok hosszára. Mivel a tényezők nagyban gyakorolnak hatást a folyamat végrehajtására, így az aktuálisan használt rendszer nagyon sok esetben okoz fennakadást az ellátásban. Mindannyian találkozhattunk már tömött sorokkal, például a napjainkban épp aktuális Sars-Cov2 védőoltások felvétele során. Az infrastruktúra javításához figyelembe kell venni további tényezőket is. Ide tartozik a páciens ráérésének, illetve a rendelési idők metszetének meghatározása, illetve a speciális vizsgálatok hosszának definiálása.

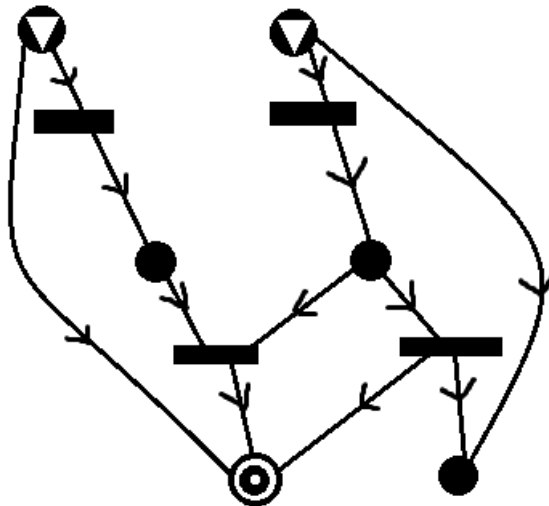
P-gráf

Az ezelőtti blokkban felvázolt tézis komplexitását és összetettségéből kifolyólag az alkalmazott alpmódszertannak a P-gráfra esett a választásom, amely a Process-graph rövidítése. A módszertan az 1990-es években fejlesztették ki, első körben vegyipari rendszerek modellezésére és optimalizálására. Alapja egy irányított, páros gráf. Felépítésének köszönhetően lehetőségünk nyílik használatával akár nagyméretű problémák megoldására is. Főbb egységeit tekintve, anyagok és műveleti egységek építik fel. Páros gráf mivoltából, ezen típusok felváltva követik egymást. Anyagok tekintetében megkülönböztetünk nyersanyagot, köztesanyagot és végterméket. A termékeket szeretnénk előállítani a folyamat során, lehetőleg a leoptimálisabban. A nyersanyagok nevéből adódóan a rendelkezésre álló alapanyagokat reprezentálják. Köztesanyagként pedig azokra a termékekre hivatkozunk, amelyek az egyes folyamatok során létrejönnek, ám ez további feldolgozás vár rá a végtermék elkészülése érdekében. Erre tökéletes példa lehet, amikor egy étel elkészítése során elkészítünk egy önmagában is fogyasztható ételt, ám ezt felhasználjuk a recept alapján. A p-gráf szükségessége a kombinatorikus tulajdonságaikban keresendő. Egy a gyakorlatban is használt általános gráf alkalmazatlan a megoldások strukturális reprezentációjához. Alapvetően tehát egy kulcsfontosságú tényező, hogy csak azon szintézisek adhatnak optimális megoldást, amelyek ezen kombinatorikus tulajdonságoknak megfelelnek. Ezen sajátosságokat axiómáknak is szokás nevezni, amelyek esetünkbe a következők:

- „Minden legyártandó termék, azaz P minden eleme szerepel a struktúrában.”
- „Egy a struktúrában szereplő anyag akkor és csak akkor nyersanyag, ha egyetlen a struktúrában szereplő műveleti egység sem állítja elő.”
- „Minden a struktúrában szereplő műveleti egység a szintézis feladatban definiált.”
- „Minden a struktúrában szereplő műveleti egységből vezet út legalább egy legyártandó termékhez.”
- „Ha egy x anyag része a struktúrának, akkor létezik a struktúrában olyan műveleti egység, amelynek x anyagot felhasználja vagy előállítja.”

Az axiómák külön tekintve elég logikusnak tűnhetnek. Azonban együttesen tekintve tökéletesen alkalmazható, a nem megfelelő felépítésű struktúrák kiszűrésére.

Az egyes anyag és műveleti egységeket élek kötik össze, ám két csúcs közt csak akkor húzható él, amennyiben a kapcsolatuk része a teljes folyamatnak. Az élek irányát pedig az határozza meg, hogy mely irányba halad a folyamat. A megoldás során a struktúra nem feltétlen használ fel minden anyagot, amely rendelkezésre áll. P-gráf struktúrát különböző matematikai algoritmust felhasználva is értelmezhetünk. Ilyen például az MSG, vagy másnéven Maximális Struktúra Generáló algoritmus, amely a gráf axiómáin alapszik. Ennek hátránya, hogy nem feltétlen a legoptimálisabb szerkezetet építi fel. Ennek kiaknázásaként beszélhetünk az SSG algoritmusról is. Ez a rekurzív folyamat, minden lehetséges struktúrát legenerál, amelyek közt megtalálható a maximális is, mint megoldás. Harmadikként pedig az én megoldásomban is használt vegyes-egész matematikai programozási modellt említeném. A korábbiaktól eltérően ez nem csak a strukturális szempontból vizsgálja a szintézis feladatot. Alapvetően ebben az esetben megkülönböztetünk folytonos és bináris változókat is, amelyek a műveleti egységeket reprezentálják. A folytonos változók értékével adható meg az egyes műveleti egység mérete, amíg a bináris az, hogy adott az adott változó részt vesz-e az optimális struktúrában, azaz, $y=1$ teljesül adott elemre, akkor az x változók értékét figyelembe vesszük a megoldásban. Adott x folytonos változó a következőképp definiálható a szakirodalom szerint: $x_i \leq y_i * U_i$. y változónk mutatja az adott bináris változót, amíg az U pedig a felsőkorlátot. Előfordulhat, hogy feladat nem ad meg U értéket, ebben az esetben az esetben tetszőleges, megfelelően nagy szám használatos. A módszer szerint a célfüggvényünk minden esetben a költségminimalizálást adja meg. Költség alatt minden beruházási költség és minden működtetés közben felmerülő költségek összegét értjük. A folytonos változót definiáló korlát mellett, megadandók anyagegyensúlykorlátok, amelyek biztosítják, hogy annyit hozzunk létre a köztesanyagokból, amely mennyiség az azt felhasználó folyamat igényeit kielégíti. Ezekon felül szükséges alsókorlát a termékek mennyiségére, illetve a rendelkezésre álló nyersanyag mennyiség szerint felsőkorát(ok). Ez a hármas biztosítja, hogy a folyamat ne akadhasson meg, illetve, hogy annyi anyaggal dolgozzunk amennyi rendelkezésre áll. Emellett persze a megfelelő mennyiségű termék előállítását is biztosítja. A következő képen egy példát láthatunk egy P-gráf struktúrára:



1. ábra: P-gráf struktúra felépítése

Időkorlátos folyamatszintézis

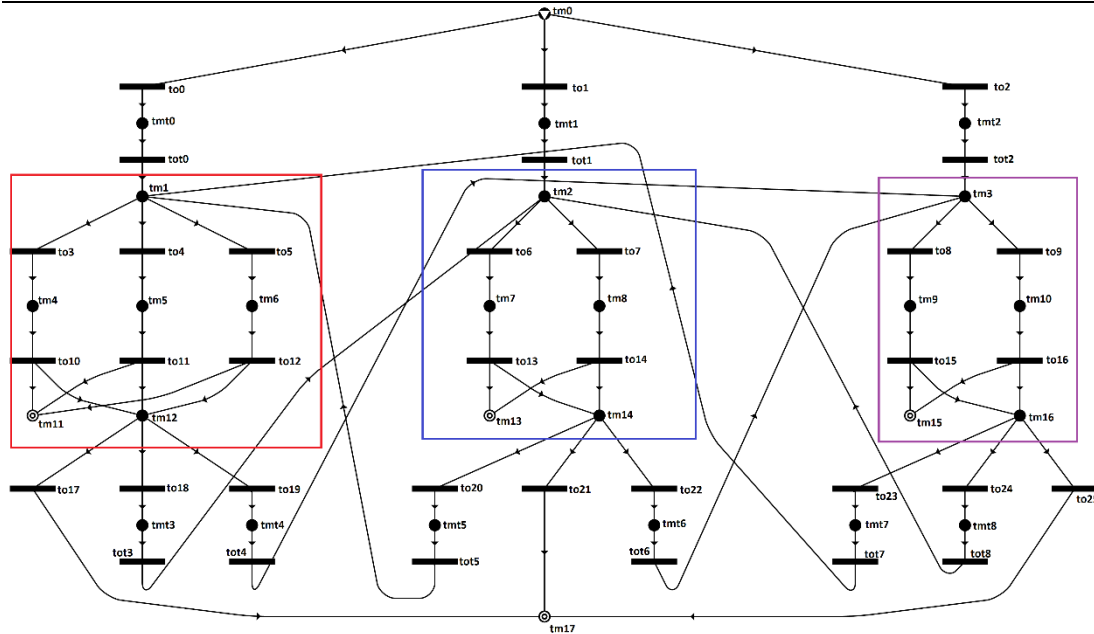
Az ütemezési feladatok azonban egy összetettebb megoldási struktúrát igényelnek. Emiatt a P-gráf önmagában nem elégíti ki elvárásainkat. Megoldás lehet más módszerek használata, mint például az S-gráf, amelyben a S az ütemezés angol megfelelőjének a schedule szónak a rövidítését takarja. Azonban egy másik lehetőség is felmerül lehetőségként. Ez a módszertan, az időkorlátos folyamatszintézis (TCPNS), amely Egyetemünk egyik neves oktatójának, Dr. Bertók Botond nevéhez fűződik. Ez a feladatosztály a P-gráf módszertan kiterjesztése, amelyet így ütemezési feladatok megoldására is alkalmassá teszi. Ebben az értelmezésben minden anyagot időként kell definiálnunk. Ennek metódusaként, létre kell hoznunk új változókat, amelyek az alapmodellel ellentétben nem csak a műveleti egységekre írandók, hanem az anyagokra is. A felső és alsó korlátjainkkal is időegységeket adhatunk meg. Emellett az egyes egységeket fix költséggel is el kell lássuk, amelyek általában egy a folyamat elvégzéséhez szükséges időt fogja megadni. Mindezek mellett szükséges néhány új korlátot is definiálnunk, hiszen ebben az esetben nem elégséges a gyártási problémákat leíró struktúra. A kibővítés során meghatározzuk, hogy az egyes tevékenység befejezésének ideje befolyásolja a következő esemény kezdési időpontját. Ez egyenlet szinten leírva a következő lesz: $O_i = (\alpha_i, \beta_i) \in O, \forall m_j$. Ez leírja, hogy az egyes műveleti egységek kezdési ideje nem lehet kisebb, mint az azt megelőző anyag típusú csúcs ideje. Az új korlátok leírják számunkra, hogy egy következő alkalom kezdésének ideje nem lehet nagyobb, mint az azt megelőző kezdési idejének és a hosszának összege. Egyenletként ez

így definiálható: $O_i = (\alpha_i, \beta_i) \in O, \forall m_j \in \beta_i : t_{m_j} \geq t_{f_i} + x_i * t_{p_i}$. Ezen egyenlet pedig kimondja, hogy minden műveleti egység idejének, illetve fix költségének idejének összege. Ezzel biztosítható, hogy az egyes alkalmak ne lehessenek átfedésben. Azonban ezen két kitétel csak abban az esetben alkalmazható, egy lineárisan egy irányban végig menő p-gráfról beszélünk, viszont az ütemezés során lehetséges, hogy visszakell térjünk egy másik ágba. Emiatt egy lazított modell is rendelkezésünkre áll. Ebben az utóbbi korlátozás tartalmaz némi plusz kitétel, amely így ebben a formában érvényes: $O_i = (\alpha_i, \beta_i) \in O, \forall m_j \in \beta_i : t_{m_j} \geq t_{o_i} + x_i * t_{p_i} + y_i * (t_b + t_{f_i}) - t_b$. Ezzel egy olyan plusz feltétel került a korlátozásba, hogy csak abban az esetben történjen meg a számítás, ha az egység benne van a megoldási struktúrában, különben 0 értéket vegyen fel. A probléma jellegét felmérve választásom az utóbbi megoldás használatára esett, így az időkorlátos folyamatszintézist alkalmaztam a munkám során.

Betegvizsgálatok ütemezése folyamathálózat-szintézis eszközökkel

A történelmi áttekintés során a legszembevetőbb problémát a betegvizsgálatok ütemezése jelentette. Kutatásaim során a P-gráf szintézist, azon belül pedig a speciális TCPNS struktúrát találtam a folyamat optimalizálásához legmegfelelőbb módszernek. Alap koncepció szerint meg kell határozzuk a vizsgálatokat, illetve azt, hogy ezekhez mennyi időegységet rendelünk. Az éles kórházi folyamatokban legtöbbször az áll fenn, hogy több rendelkezésre állási időszávot kapunk, amelyek megadják mikor érhetjük el az adott vizsgálatot. Mivel egy pácienshez több vizsgálatot is rendelhet a kezelőorvosa és minden rendeléshez tartozhat több időpont, így értelemszerűvé vált, hogy a megoldás során automatikus modellgenerálásra feltétlen szükség lesz. Az előző pontban bemutatott p-gráf struktúra egy teljesen általános felépítés. Ezzel szemben a betegütemezésében egy sokkal speciálisabb, sajátos szisztéma szerint épül fel. A vizuális felvázolás esetén már, akár ránézésre látható, hogy ütemezési problémáról van szó. Ennek oka, hogy a folyamat felvázolása során jól elkülöníthető részre osztja az egyes vizsgálatokhoz tartozó részegységeket. A P-gráf alapmodellben találhatunk nyersanyagot, köztesanyagot, termékeket, illetve műveleti egységeket. Mivel az időoptimalizálás esetében nem beszélhetünk egyértelműen kézzel fogható dolgokról így ezek egy kis magyarázatra szorulnak. A nyersanyag reprezentálja számunkra a folyamat kezdetét. Értendő ez alatt például, hogy megérkezünk az ellátó intézménybe. Ez a művelet során egy számértéket fog kapni, amely megadja, hogy a vizsgálati sor milyen időpontban kezdődik majd el. Mivel az alap séma egy páros gráf, így annak tulajdonságait figyelembe véve ezt egy műveleti egység követi. Ahogy az eredeti modell során, itt is ezen típusú egységek reprezentálják az egyes műveleteket. Az eltérés abban nyilvánul meg, hogy nem valamilyen gyártói folyamat megy végbe itt, hanem esetünkben egy egészségügyi vizsgálat. Ezeket követi újra egy anyag, amely a köztes anyag lesz. Ez mutatja meg, hogy a megtörtént kivizsgálás után milyen időpontnál tartunk. Ezen felül megadja, amennyiben nem az utolsó lépésnél tartunk azt is, hogy mikor kezdődhet az új lépés, avagy a másik helyszín megközelítése. Az alapvető felépítést a következő ábrával szeretném bemutatni:

Betegvizsgálatok ütemezése folyamathálózat-szintézis eszközökkel



2. ábra: TCPNS struktúra 3 vizsgálat esetén

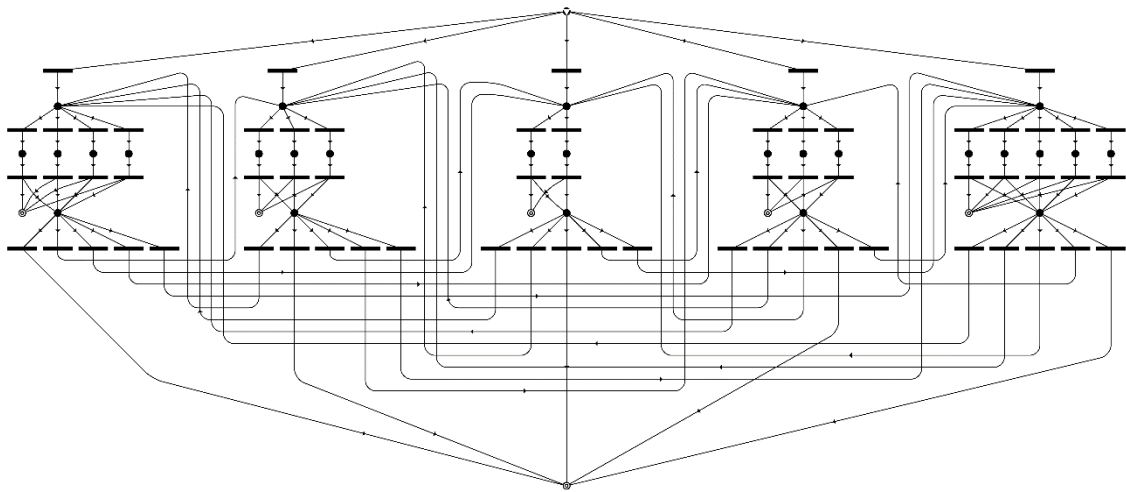
A folyamat illusztráláshoz egy olyan példát választottam, amely során 3 vizsgálatot írt elő az orvos a betegnek. Ezeket az emberközelibb megfogalmazás kedvéért nevezzük el a következőképp: vérévétel, CT, illetve röntgen. A vérévételhez tartozó piros négyszögben láthatjuk, hogy 3 rendelkezésre álló időszáv van az előjegyzési naptár szerint. Azonban a CT és röntgenek, ahogy a kék és lila blokk mutatja 2-2 időszávot adnak meg számunkra. Az ábrán látható, az előzőekben már említett egyedi felépítés. Maga a folyamat a nyersanyagból indul, amely meghatározza a megoldás struktúrában a kezdeti időpontot. Ez elágazik esetünkben 3 részre. Ez a három színes blokk az egyes vizsgálatokat jelzik. Mindegyik egységbe megy egy él, hiszen nem tudható melyikkel fogja kezdeni a megoldó. Az egyes modulokban további elágazásokat figyelhetünk meg. Ezek adják meg a konkrét vizsgálathoz több időpont tartozik és bármelyik felvehető a megoldás során. Ezeken belül, ha az első, piros egységet tekintjük példaként a to_3 - to_4 - to_5 műveleti egységek lesznek a vizsgálati folyamatok, azaz megadják mikor milyen intervallumban mehetünk be és milyen hosszú ideig fog tartani a folyamat. Minden blokkhoz tartozik egy-egy termék is, hiszen egy rendelésen csak egyszer vehetünk részt, így mindegyik egység „előállít” egy időpontot mely megadja, hogy mikor értünk annak a végére. Látható minden színes négyszög alatt n_i , amely esetünkben 3 darab műveleti egységet. Ezek képviselik az egyes helyszínek közti átérést. Hármasonként lebontva az n_i indexű egység mutat az n_i indexű egységbe. Mindegyik közt azon sorszámú egység, amely az adott blokk sorszáma szabályszerűen a végtermékbe mutat. Ezek mutatják meg, hogy adott vizsgálat után mindegyik másik vizsgálat is következhet, vagy abban az esetben, ha utolsónak választja a megoldó szoftver, akkor pedig a teljes folyamatot lezáró végtermék lesz a következő állomás. A p-gráf mivoltából kifolyólag minden elemnek van egy alsó és felső korlátja. Mivel nem beszélhetünk rendelkezésre álló alapanyagmennyiségről, illetve kötelezően előállítandó mennyiségű termékről, így nagyrészt 0 és egy kellően nagy szám közt lesznek az egyes elemek értékei. Ezen állítás alól kivételt képeznek az egy blokkok első során képező műveleti egységek. Ábránkon látható példánk esetében ez konkrétan a to_3 - to_9 közti rész. Ennek oka, hogy mivel ezek reprezentálják az egyes

Betegvizsgálatok ütemezése folyamathálózat-szintézis eszközökkel

rendeléseket, így ezek alsó korlátja az intervallum nyitási ideje, míg felső korlátja a zárás. További feltételként meg kell határozzuk, hogy minden vizsgálatot egyszer választhasson a szoftver, illetve rendelésenként csak egy időpontot és természetesen az átérések közül is. Ehhez a t_{00} - t_{0n} közt, blokkonként az első műveleti egységekre, illetve rendelésenként az átérést jelentő egységekre szükséges egy-egy anyagegyensúly korlát. Ahogy már az előző pontban említettem, ez az adott egységhez tartozó bináris változókkal írandó fel és lényege, hogy egységenként $\sum y_i = 1$, azaz mindenképp csak egyet választhasson be a megoldó mindegyikből, ezzel elkerülve az esetleges kellemetlenségeket.

Az automatikus modell generálás során n vizsgálat és $m_0 - m_n$ egyenként különböző adatokra is megfelelő modellt generáljon. Attól függően, hogy az n , illetve m értékeink mekkora számot vesznek fel, annál összetettebb vagy könnyebb a modell a felépítése. Legegyszerűbb eset például amikor $n=1$, hiszen ebben az esetben teljesen mindegy melyiket választja ki a megoldó. Amennyiben az n értéke 2, azaz 2 vizsgálatot is kell sorrendbe állítania a szoftvernek már egy jóval bonyolultabb felállást kell figyelembe venni.

Munkám során arra törekedtem, hogy a lehető legegyszerűbb felhasználási módot érjem el, így a vizuális reprezentációt nem tartottam fontosnak a megoldó szoftver elkészítésénél. A bonyolult megértést jól mutatja, hogy ha már 5 vizsgálatot szeretnék futtatni, egy nagyon komplex, első ránézésre átláthatatlan ábrát kapunk:



3. ábra: TCPNS struktúra 5 vizsgálat esetén

A megoldó szoftver készítése során egy a Google által készített sok oldalú, nyíltforráskódú program könyvtára esett a választásom, amely az OrTools nevet viseli. Előnye, hogy széles nyelvi specifikációval rendelkezik. Mind C alapú nyelvekhez találhatunk használható verziót, mind pedig a széles körben elterjedt Java és Python nyelvekhez is. Másik nagy előnye, hogy a Google saját megoldója mellett, több harmadik féltől származó forrást is használhatunk. Mivel a felhasznált módszertan alapjául szolgáló p-gráf struktúrában a MILP, vagyis Mixed-integer linear programming, ami magyarul Vegyes-egész lineáris programozási leírást használtam, így a megoldó szoftver készítése során a Lineáris megoldó közül választhattam. Végül a C# környezett mellett döntöttem, hiszen abban szereztem a legtöbb tapasztalatot a korábbiakban, illetve a .Net keretrendszer segítségével, a továbbfejlesztési lehetőségek

Betegvizsgálatok ütemezése folyamathálózat-szintézis eszközökkel

közt megjelenhet a webalapú alkalmazás készítése is, amely alapja így az általam megírt szoftver lehet. Megoldóként az SCIP-re esett a választás, amely jelenleg az egyik leggyorsabb nem kereskedelmi célú megoldó. Előnye, hogy kifejezetten a C alapú nyelvekhez készült, amely nagyon megkönnyítette a vele való munkát. Emellett a gyorsabb, ám kereskedelmi célú megoldásokkal ellentétben használata teljesen szabad és ingyenes.

A Módszertan Validálása

A tesztéről általánosan

A validálás során fontos szempont volt, hogy az alkalmazott célfüggvény, azaz a folyamat kezdete és vége közt eltelt idő minimalizálása értelmében a páciens a lehető legkevesebb időt töltse az intézményben. Mivel az elkészített szoftver megoldása során nagy figyelmet fordítottam arra, hogy dinamikusan kezelve, a beírt adatok alapján automatikusan generálja a modellt, így fontosnak tartottam, hogy a lehető legtöbb lehető legtöbb esetet megvizsgáljam. A felépítéséből adódóan szükséges, hogy a megoldó szoftver megfelelő eredményt különböző vizsgálat szám esetén is, illetve, ha ezekhez különböző számú rendelési idő tartozik.

Első validációs tesztet – 1 vizsgálat 2 időpont

Az első tesztet során, a legegyszerűbb értelmezhető esetet vettem alapul. Itt egy rendelést adunk a programnak, amelyhez kettő különböző rendelési idő tartozik. Természetesen mivel a vizsgálat azonos, így a fix idejük is azonos lesz. Ezenkívül ebben a változatban nem lesz egyetlen átérési idő sem. A példa során a következő adatokat adtam át a megoldónak:

Fix idő (perc)	Ablak nyit	Ablak zár
7	8:00	8:10
7	9:00	9:10

4. ábra: Példa adatok 1 vizsgálat esetén

Két teljesen más órában lévő intervallum.

Futtatás után a következő eredményt kapjuk:

```
to0= 32520  
tot0= 32520  
to2= 32520  
to4= 32520  
to5= 32940  
Intézményben töltött idő: 420
```

5. ábra: Futási eredmény 1 vizsgálat esetén

Ahogy látható az ábrán, a megoldásban, 32520 másodperc a kezdés időpontja. Ez átszámítva 09:02. Azaz a második időintervallumba adott nekünk időpontot a rendszer. Intézményben töltött idő 420 másodperc, ami 7 percnak felel meg. Ez megegyezik a vizsgálat hosszával, hiszen most az egy helyszín miatt nem volt átérési időértékünk. A rendelkezésre álló adatok alapján, a használt célfüggvénnyel bármelyik időszávot választhatja a szoftver, hiszen az nem befolyásolja a kórházban töltött időt. Mivel az nyitvatartási idő hosszabb a maga tíz percével, mint a vizsgálat hossza, így a kapott eredmény, hogy nem 09:00-ra adja az időpontot, nem számít hibának.

Második validációs tesztet – három vizsgálat, három – kettő – kettő időszávval

Következőnek, egy az automatikus modellgenerálás szempontjából összetettebb példát vittem be a rendszerbe. A célfüggvény továbbra is a minimális intézményben töltött időt célozza meg. Ebben az esetben is a korábbiakban használt adatok kiegészített verziójával dolgoztam. Ezek az alábbi táblázatban találhatóak, az egyes vizsgálatok különböző színekkel való jelölésével:

Fix idő (perc)	Ablak nyit	Ablak zár
7	8:00	8:10
7	9:00	9:10
7	10:00	10:10
30	9:20	10:00
30	10:20	11:20
20	10:00	11:00
20	11:30	12:30

6. ábra: Példa adatok 3 vizsgálat esetén

Ezek mellett az átérési idők is képbe kerülnek ettől a példától kezdődően. Ezen értékeket a következő ábra tartalmazza:

Kiinduló rendelés - cél rendelés	Fix idő (perc)
1 - 2	5
1 - 3	25
2 - 1	20
2 - 3	1
3 - 1	30
3 - 2	5

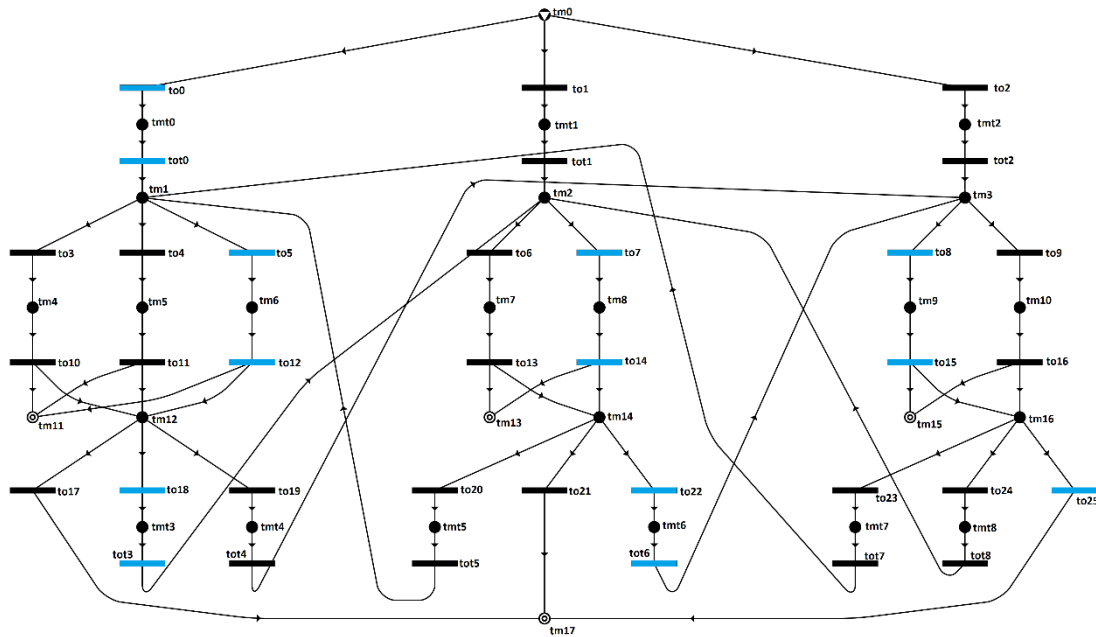
7. ábra: Átérési adatok 3 vizsgálat esetén

A feladat lefuttatása után a következő eredményt kaptuk:

```
to0= 36480
tot0= 36480
to5= 36480
to7= 37200
to8= 39060
to12= 36480
to14= 37200
to15= 39060
to18= 36900
to22= 39000
to25= 40260
tot3= 37200
tot6= 39060
Intézményben töltött idő: 3780
```

8. ábra: Futtatási eredmény 3 vizsgálat esetén

Az eredmény 3780 másodperc, ami átváltva, 1 óra és 3 perc. Ennyi időt töltünk a kórházban, ha a fent látható lista szerint haladva megyünk a vizsgálatok közt. Részletezve, az első vizsgálattal kezdjük a sort 10:08-kor. Ennek vége 10:15-kor van, innen a második vizsgálatra kell menjünk. Ide 10:20-kor érkezünk. Mivel az egyik időszáv ekkor nyit ezen a helyen, ezért itt egyből várakozás nélkül be is jutunk. Vége 10:50-kor lesz. Ezután ellátogatunk a harmadik helyszínre is, ami egy percet vesz igénybe, így 10:51-kor meg is érezzük. Mivel itt is egy épp nyitott intervallumban érkezünk, ezért rögtön el is végzik rajtunk a kért vizsgálatot, ami 20 percet vesz igénybe. 11:11-kor pedig lezárul az utolsó vizsgálatunk ezzel pedig a jelenlegi folyamatunk is. Az optimálisnak vélt folyamat, strukturális felépítése a következő kép alakul:



9. ábra: Megoldási struktúra 3 vizsgálat esetén

A kék színnel jelölt műveleti egységek kerülnek bele a folyamatba, így vizuálisan is végig követhető a folyamat sorrendje.

Harmadik validációs teszteset – három vizsgálat, három – kettő – kettő időszávvval

Az előző tesztesettel megegyező struktúrát követ a jelenlegi validációs teszt is, ám az adatok merőben mások lesznek. Hasonlóan a korábbiakhoz egy-egy táblázat tartalmazza a nyitvatartási időket, a vizsgálatok fix idejét, azaz hosszát, illetve a

A Módszertan Validálása

helyszínek közti átérési időket is:

Fix idő (perc)	Ablak nyit	Ablak zár
15	08:15	08:45
15	11:30	12:00
15	13:00	13:30
5	09:10	09:30
5	10:00	10:15
45	15:00	16:00
45	17:30	18:30

10. ábra: Példa adatok 3 vizsgálat esetén

Kiinduló rendelés - cél rendelés	Fix idő (perc)
1 - 2	0
1 - 3	15
2 - 1	10
2 - 3	30
3 - 1	20
3 - 2	5

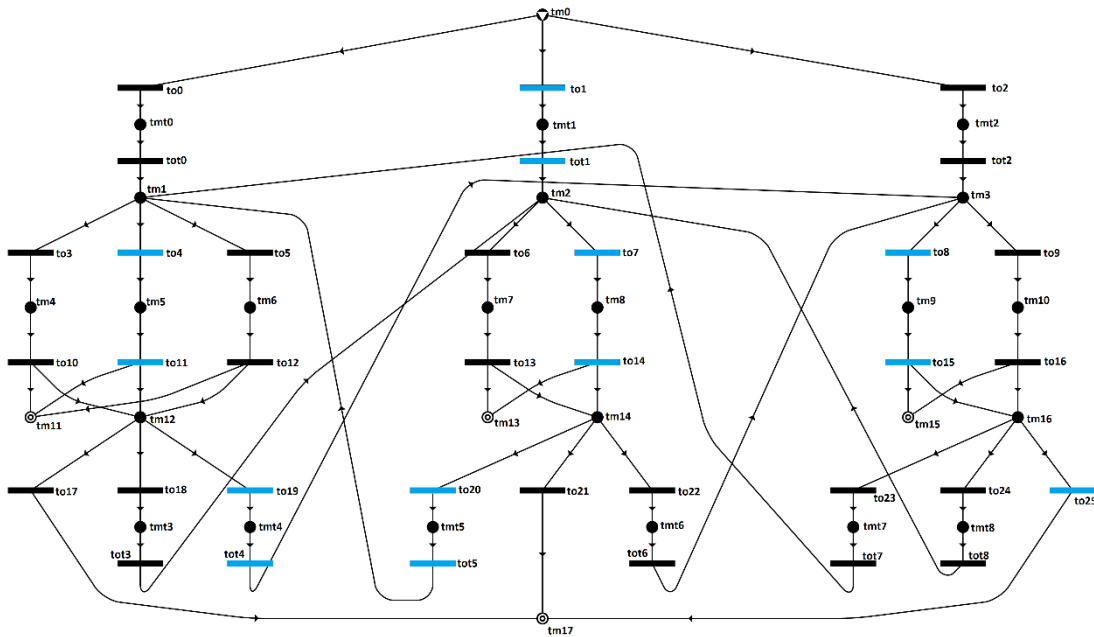
11. ábra: Átérési adatok 3 vizsgálat esetén

Látható, hogy merőben más tesztesetről beszélünk, az adatok szintjét tekintve. Ennek a programba való betáplálása után, a rendszer a következő eredményt tárja elénk:

```
to1= 36900
tot1= 36900
to4= 37800
to7= 36900
to8= 54000
to11= 41400
to14= 36900
to15= 54000
to19= 42300
to20= 37200
to25= 56700
tot4= 50400
tot5= 37800
Intézményben töltött idő: 19800
```

12. ábra: Futtatási eredmény 3 vizsgálat esetén

Eredményünk 19800 másodperc, azaz 5 óra 30 perc. Szemmel láthatóan nagyobb időintervallumról beszélünk, mint a korábbiakban bárhol. Ennek oka a rendelési idők túlzott szétdaraboltsága. A lefuttatott folyamat a kettes számú vizsgálattal indul, 10:15-kor, ennek hossza 5 perc, így 10:20-kor már indulhatunk is a következő vizsgálatra, ami az egyes sorszámú. Ide az átérési idő, szintén 5 perc. Azonban mivel nincs nyitott vizsgálati ablak, így 1 óra és 5 perces várakozás után, 11:30 kezdődik meg a következő 15 perces esemény. Innen 11:45-kor elindulhatunk, egy 15 perces úton a harmadik helyszínre, ahol szintén hosszabb várakozás után kezdetét veszi az utolsó, 45 perces folyamat. A kórházban töltött időnk, 15:45-kor fejeződik be. A folyamat a képen látható struktúra alapján épül fel, ahol szintén a színes egységek jelzik a folyamatban résztvevő egységeket:



13. ábra: Megoldási struktúra 3 vizsgálat esetén

Negyedik validációs teszteset – három vizsgálat, három – kettő – kettő időszávvval

A teljes bizonyosság érdekében, újabb 3 vizsgálatos esetet futtattam le a rendszeren. Az adatok nagyon hasonlóak az előzőhöz és ismét táblázatokban foglaltam őket össze:

Fix idő (perc)	Ablak nyit	Ablak zár
12	11:30	11:45
12	12:00	12:30
12	12:45	13:15
10	08:00	08:20
10	09:15	09:40
15	09:30	10:00
15	11:10	11:40

14. ábra: Példa adatok 3 vizsgálat esetén

Kiinduló rendelés - cél rendelés	Fix idő (perc)
1 - 2	5
1 - 3	15
2 - 1	3
2 - 3	20
3 - 1	10
3 - 2	15

15. ábra: Átérési adatok 3 vizsgálat esetén

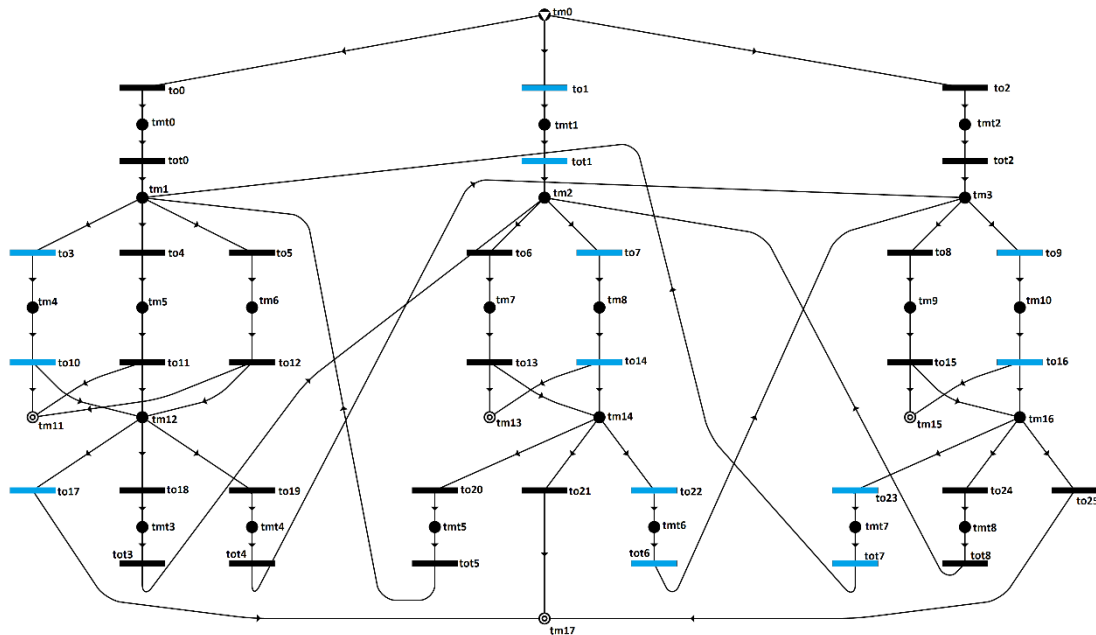
Ahogy lefuttatjuk a folyamatot, a program a következő eredményt adja át nekünk:

```
to1= 34800
tot1= 34800
to3= 41700
to7= 34800
to9= 36600
to10= 41700
to14= 34800
to16= 40200
to17= 42420
to22= 35400
to23= 41100
tot6= 36600
tot7= 41700
Intézményben töltött idő: 7620
```

16. ábra: Futtatási eredmény 3 vizsgálat esetén

7620 másodpercet, azaz 2 óra, illetve percet tölt páciensünk az épületben. A második vizsgálatlal kezdi a folyamatot 09:40-kor, a rendelőben 10 percet töltve. Ezután egy 20 perces séta után 10:10 perckor megérkezik a hármas számú rendelésre. Itt az időszávból kifolyólag kénytelen 11:10 percig várakozni. Ekkor egy 15 perces vizsgálat után elindulhat az utolsó vizsgálatra, ami az egyes számú lesz. Ide 10 perc alatt érkezhethet meg és egyből 11:35-kor már kezdődhet is a vizsgálat. Az itt eltöltött 12 perc után le is zárul

a folyamat és elhagyhatjuk az intézményt. A megoldási struktúra az előzőkéhez hasonlóan egy p-gráfban látható, ahol a színes egységek jelzik a kiválasztott elemeket:



17. ábra: Megoldási struktúra 3 vizsgálat esetén

Ötödik validációs tesztet – három vizsgálat, három – kettő – kettő időszákkal

Egy utolsó háromvizsgálatos példával szeretném biztosítani a működés sikerességét. Ehhez a korábbiakhoz hasonlóan egy táblázatot készítettem a kezdeti értékekről, amely tartalmazza a vizsgálatok hosszait, a nyitási és a zárási időpontokat, illetve percben meghatározva a helyszínek közti távolságokat:

Fix idő (perc)	Ablak nyit	Ablak zár
8	08:30	08:45
8	09:45	10:00
8	11:15	11:30
15	09:00	09:30
15	13:30	13:45
5	10:45	11:15
5	14:00	14:30

18. ábra: Példa adatok 3 vizsgálat esetén

Kiinduló rendelés - cél rendelés	Fix idő (perc)
1 - 2	20
1 - 3	30
2 - 1	5
2 - 3	2
3 - 1	15
3 - 2	10

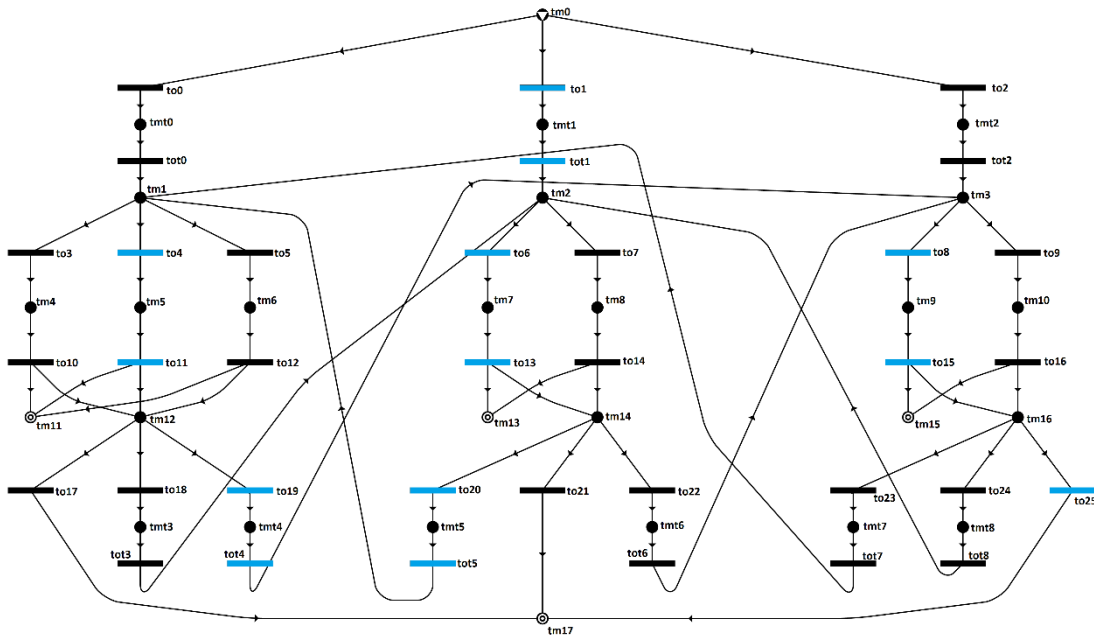
19. ábra: Átérési adatok 3 vizsgálat esetén

A szoftver az automatikus modellgenerálást és az optimalizálási folyamat lefutása után a következő eredménnyel szolgált:

```
to1= 34200
tot1= 34200
to4= 35400
to6= 34200
to8= 38700
to11= 35400
to13= 34200
to15= 38700
to19= 35880
to20= 35100
to25= 39000
tot4= 38700
tot5= 35400
Intézményben töltött idő: 4800
```

20. ábra: Megoldási adatok 3 vizsgálat esetén

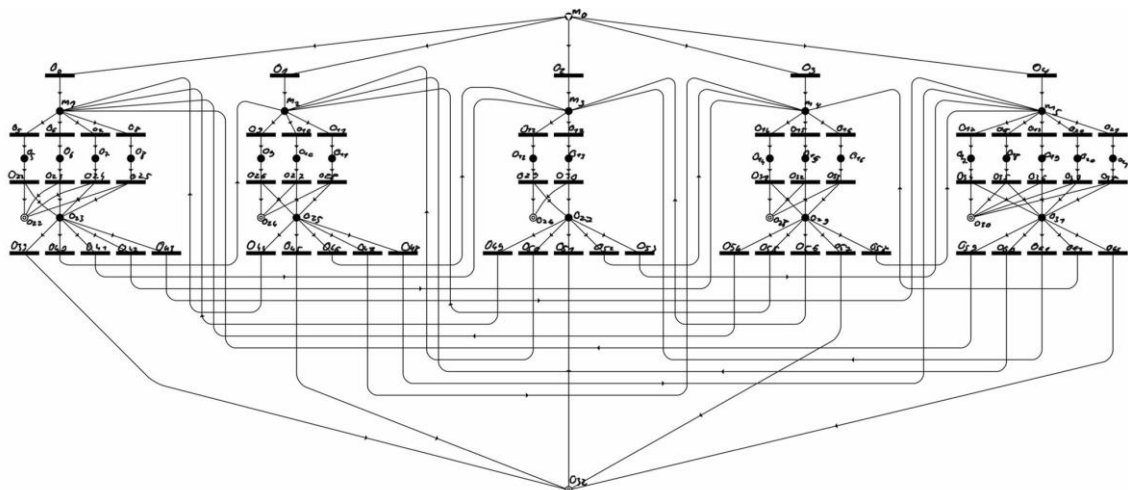
A kapott, 4800 másodperces célfüggvényérték arról árulkodik nekünk, hogy 1 óra 20 percet kell a kórházban töltenünk a szekvencia során. Maga a folyamat indulása a kettes vizsgálatnál történik, 09:30-kor. Ez 15 percet vesz igénybe, ezután 5 percet vesz igénybe az átérésünk a következő helyszínre, ahova így 09:50-kor érkezünk. Mivel itt csak 10:00-kor kezdődik a rendelés, ezért 10 perc várakozásra kényszerülünk. Miután ez és maga a 8 percet igénybe vevő vizsgálat is lezajlott, 10:08-kor elindulhatunk a hosszabb, 30 perces utunkra, amely a hármas számú helyszínre vezet. Hasonlóan az egyes számúhoz, itt is szükséges várunk, de itt csak 7 percet. A 10:45-től 10:50-ig tartó rövid procedúra után pedig zárjuk a folyamatunkat. Szemléltetésképp ezen szekvenciát is ábrázoltam p-gráf struktúrában, ahol a színes blokkok jelzik a választott egységeket:



21. ábra: Megoldási struktúra 3 vizsgálat esetén

Hatodik validációs tesztet – öt vizsgálat, négy – három – kettő – három - öt időszávvval

Az utolsó két esetben bonyolultabb felépítésben is tesztelni kívánom a megoldó szoftverem működését. Emiatt egy ötvizsgálatos szekvenciát fogok bevinni a programba, amely a következőképp épüfel:



22. ábra: P-gráf 5 vizsgálat esetén

Látható, hogy más-más időszáv mennyiséggel dolgozom az egyes vizsgálatok esetén, ezzel is nehezítve a feladatot. Első körben a következő táblázatnak megfelelő adatokkal próbáltam ki a programot:

Fix idő (perc)	Ablak nyit	Ablak zár
13	09:45	10:15
13	11:00	11:30
13	12:15	12:45
13	14:00	14:30
5	08:00	08:15
5	10:30	10:40
5	11:20	11:30
15	08:30	09:15
15	11:30	12:15
8	09:45	10:00
8	11:20	11:35
8	12:15	13:00
20	08:00	09:00
20	09:30	10:30
20	10:15	10:45
20	13:00	14:00
20	15:30	16:00

23. ábra: Példa adatok 5 vizsgálat esetén

Szemmel láthatóan nagyobb mennyiségű adattal rendelkezünk. Ebből kifolyólag sokkal jobban felderíthető az általam készített megoldó szoftver generálási képessége. Az előzőkhez hasonlóan most is a kórházban töltött időt szeretnénk minimalizálni. Miután átadtunk mindent a programnak, a következő eredményt kaptuk:

```

to3= 35880
tot3= 35880
to6= 39060
to10= 38400
to13= 41580
to14= 35880
to18= 36480
to23= 39600
to27= 38400
to30= 41580
to31= 35880
to35= 36480
to41= 40380
to44= 38700
to51= 42480
to58= 36360
to60= 37680
tot6= 41580
tot9= 39060
tot20= 36480
tot22= 38400
Intézményben töltött idő: 6600

```

24. ábra: Megoldási adatok 5 vizsgálat esetén

A vártakkal ellentétben nem nőtt meg jelentősen a teljes idő, hiszen, 6600 másodpercet kaptunk, ami 1 óra és 50 perc. Ez köszönhető az időszávok, a vizsgálati hosszok és az átérési idők jól megválasztásának. Az adatok alapján ezt a szekvenciát a négyes sorszámú vizsgálatlal indítjuk, 09:58-kor. A 8 perces művelet után, a 2 percre lévő ötös vizsgálat következik, amely érkezés után, 10:08-kor meg is kezdhető. A procedura 20 perc hosszúságú, ezután következik egy 12 perces út a hármas vizsgálatához. Itt azonnal, 10:40-kor kezdetét veheti a rövid, 5 perces tevékenység. 6 perc átállás és 8 perc várakozás után kezdetét veszi az egyes rendelés. Itt 13 percet tölt el páciensünk. Ezután 20 perc alatt átérünk az ütemezésünk utolsó pontjához, a hármas számúhoz, amely azonnal kezdetét is veheti. Az egész folyamat pedig 11:48-kor zárul. A feladat összetettsége miatt a korábban már megismert módszertő, amikor is a p-gráf struktúrában színekkel jelöltem a résztvevő elemeket, eltekintenek.

Hetedik validációs teszteset – három vizsgálat, három – kettő – kettő időszávval

Végezetül, egy újabb az előzőhöz hasonló felépítésű ütemezési feladatot is lefuttattam. Ennek adatait is egy táblázat segítségével szeretném szemléltetni:

Fix idő (perc)	Ablak nyit	Ablak zár
7	08:30	08:45
7	09:15	09:30
7	12:00	12:30
7	14:45	15:00
15	09:00	09:30
15	10:30	11:15
15	15:40	16:10
25	08:00	09:00
25	11:30	12:30
12	08:15	09:00
12	10:15	11:00
12	12:00	12:45
20	08:00	08:15
10	09:30	09:45
10	11:00	11:15
10	13:45	14:00
10	14:00	14:15

25. ábra: Példa adatok 5 vizsgálat esetén

Kiinduló rendelés - cél rendelés	Fix idő (perc)
1 - 2	3
1 - 3	5
1 - 4	15
1 - 5	10
2 - 1	25
2 - 3	13
2 - 4	45
2 - 5	0
3 - 1	6
3 - 2	2
3 - 4	1
3 - 5	12
4 - 1	10
4 - 2	7
4 - 3	5
4 - 5	3
5 - 1	15
5 - 3	20
5 - 1	25
5 - 3	5

26. ábra: Átérési adatok 5 vizsgálat esetén

A program a betáplált adatokból, automatikus modellgenerálás és optimális megoldás keresés után, a következő eredményt adta:

```
to2= 30720
tot2= 30720
to6= 33600
to9= 34200
to12= 30720
to14= 32280
to18= 35100
to23= 33600
to26= 34200
to29= 30720
to31= 32280
to35= 35100
to40= 34020
to48= 35100
to52= 32220
to54= 33000
to63= 35700
tot5= 34200
tot12= 35100
tot15= 32280
tot17= 33600
Intézményben töltött idő: 4980
```

27. ábra: Megoldási adatok 5 vizsgálat esetén

4980 másodpercet, azaz 1 óra 20 percet vesz igénybe a folyamat. Korábban már a 3 vizsgálatból álló folyamat esetén is találkozhattunk ennél magasabb időeredménnyel. A jól megválasztott időszávok, illetve a nagyobb választási lehetőséget adó bemenet esetén, több rendelésre való küldés esetén is érhetünk el alacsonyabb végeredményt. Jelenlegi ütemezésünk, a hármas vizsgálatnál kezdődik 08:32-kor, ami 25 percet vesz igénybe. Ezek után vár ránk egy rövid, 1 perces séta a négyes rendelés helyszínére. Itt már az érkezés pillanatában, 08:58-kor megkezdődhet az ellátás. Egy rövid 12 perces vizsgálat után, újabb 10 perces átállás következik az egyes vizsgálatra. Itt újra azonnali fogadás történik, így a 7 perces folyamat érkezéskor, 09:20-kor meg is kezdődhet. Következő, kettes számú vizsgálatához 3 perc utazásra van szükségünk. 09:30-kor érkezünk a helyszínre, ahova belépve már meg is kezdődik a panasz felderítése, amely 15 percet vesz igénybe. Mivel az utolsó vizsgálatunk, az ötös számú 0 percre van a jelenlegi helyzetünktől, tehát feltehetőleg egy vizsgálóban történik a kettő, ezért azonnal, 09:45-kor már el is kezdődhet a 10 perces folyamat. Ezzel a 10 perc híján 1,5 órás ügymenet

végére értünk. Hasonlóan az előzőhöz, a példa összetettsége miatt a vizuális megjelenítéstől eltekintek.

Irodalomjegyzék

- I. Dr. Bertók Botond – Dr. Kovács Zoltán 2011. *Gyártórendszerek modellezése*. Typotex Kiadó, Veszprém
- II. Károly Kalauz – Zoltán Süle – Botond Bertók – Ferenc Friedler 2012. *Extending Process-Network Synthesis Algorithms with Time Bounds for Supply Network Design*. AIDIC Servizi S.r.l.
- III. Reza Alizadeha – Javad Rezaeiana – Mehdi Abedib – Raymond Chiong 2019. *A modified genetic algorithm for non-emergency outpatient appointment scheduling with highly demanded medical services considering patient priorities*. Elsevier Ltd.
- IV. <https://www.scipopt.org/> (megtekintve: 2021.11.14.)
- V. <https://developers.google.com/optimization> (megtekintve: 2021.11.14.)

Összegzés

A munkám során egy olyan mindennapokban is felmerülő probléma elméleti megoldásával foglalkozhattam, ami szinte minden ember életére kihatással van. A probléma és a téma irodalmának feltárása és korábbi ismeretek mélyebb megismerése során rádöbbsentem, hogy minden eredmény, ami az egészségügy hatékonyabb működését eddig segítette, illetve a jövőben ezt célozza meg, nagyban hozzájárul mind a dolgozók eredményes munkájához, mind pedig ahhoz, hogy a kórházakba látogató páciensek intézményben töltött idejét a lehető legrövidebbre korlátozzák. Ezek a tevékenységek mind a jövőbeni hatékonyabb egészségügyi ellátást tűzték ki maguk elé, ezzel téve egyre könnyebbé életünket.

A dolgozatom elkészítése folyamán a programozás egy olyan ágával ismerkedhettem meg, amely merőben más gondolkodásmódot igényelt, mint a korábbi tevékenységeim. A folyamat során módom nyílt egy új területen fejlődni, amely a későbbiekben is előnyömre válhat.

Úgy vélem a későbbiekben az elkészített munkám továbbfejleszthető egy, a gyakorlatban is könnyen használható asztali szoftverré, vagy webalapú alkalmazássá.