

A kutatás célkitűzése a Petri-háló és a produkciós háló közötti kapcsolat feltárásának segítségével olyan hatékony analízis és optimalizálási módszerek kidolgozása volt, melyek eszközként szolgálnak a bizonyítottan helyes működésű informatikai rendszertervezés, vagyis a tervezéshez kapcsolódó formális validáció és verifikáció elvégzéséhez.

### ***Petri-háló és produkciós háló közötti kapcsolat***

A kutatás első felében elméleti oldalról alapoztuk meg a későbbiekben végzendő tevékenységet [6,8,19,21-25]: kidolgoztuk a Petri-háló és produkciós háló egységes szemléletű leírását. Az e téren végzett kutatásokról összefoglaló jelleggel egy folyóiratcikk is íródott [36].

Fő eredménynek tekinthető, hogy miközben a két formalizmus gond nélkül egységesíthető, behatároltuk azokat a strukturális jellegű megkötéseket, amelyeket a Petri-háló esetében fel kell oldani annak érdekében, hogy a produkciós háló optimalizálására kidolgozott módszerek hasznosíthatóak legyenek (a Petri-háló kezdő tokeneloszlásának transzformációja, illetve az úgynevezett katalizátor probléma megoldása) [6,8,19,21-25,36].

Ezáltal elvileg tetszőleges Petri-háló optimalizálására alkalmassá válik a produkciós háló optimalizálására kidolgozott megoldástér szűkítő és optimalizálást gyorsító algoritmuscsalád (gyorsított branch and bound). Ugyanakkor, a produkciós hálónál a probléma megoldására kidolgozott hatékony algoritmusok *nem egész megoldást* keresnek, amely azonban elvárt a Petri-háló esetében, így a PNS algoritmusok csak azzal a kiegészítéssel adaptálhatók közvetlenül a Petri-háló alapú modellek optimalizálására, hogy az optimalizálási fázisban egész megoldásokat keressünk [36].

A produkciós háló esetében az algoritmusok az optimális megoldás *megvalósítását* nem garantálják, amely azonban elengedhetetlen a Petri-háló alapú modellek esetében, központi kérdés volt az optimalizálási fázisban kapott megoldás tüzelhetőségének vizsgálata, valamint az ennek megfelelő trajektória megadása, melyet külön fázisban ellenőriztünk, illetve adtunk meg [19,21-25,36].

Mivel a tüzelhetőségi vizsgálatok elvégzése a tradicionális módszerekkel kombinációs robbanással fenyeget, szimbolikus minimalizálási technikák bevezetésével e vizsgálatokat kompakt formában sikerült elvégezni [21-25]: az adott megoldás lényegében a Petri-háló, mint nem determinisztikus véges automata állapotátmeneti relációját építi fel bináris döntési diagramok (BDD) segítségével, amelyből egy nem determinisztikus – determinisztikus automata transzformáció után a tüzelhetőség az adott kezdőállapotból elérhető állapotok meghatározásával dönthető el. Ez az állapotátmeneti függvény (Petri-háló tüzelései) tranzitív lezártjának meghatározásával valósítható meg, amelyre hatékony szimbolikus technikák léteznek.

A kidolgozott algoritmusban az informatikai rendszerek tipikus verifikációs feladatait a tüzelhetőségi vizsgálatba építettük be: a SPIN modellellenőrző eszköz a modellezett

informatikai rendszerek verifikációját támogatja temporális logikai feltételként megfogalmazva [32]. Adott temporális logikai feltétel ellenőrzésével a SPIN a verifikációs feladatok vizsgálata mellett időbeli követelmények vizsgálatát is lehetővé teszi (pl. az y folyamat sohasem kezdődhet el x folyamat előtt) mind a diagnosztikai feladatok, mind a termelési folyamatok vizsgálatában [19,21-25].

A 2003-ban kiadott, a SPIN modellellenőrző eszköz legújabb verziójának segítségével az optimalizálásért felelős Branch-and-Bound módszer feltételei ugyanakkor a verifikációs feltételbe (mely egy lineáris temporális logikai kifejezés) ágyazhatóak [28]. Ez a megoldás az optimális, a verifikációs követelményeknek megfelelő útvonal meghatározását az állapottér bejárása közben dinamikusan végzi (szemben az előző módszerrel, mely először statikusan optimalizál, majd utána vizsgálja a megoldás megfelelőségét verifikációs szempontból). Ugyanezt a megoldást használja a gráftranszformációs rendszerek optimalizálására adott megoldásunk [31], ahol a költségfaktorok szerepét az egyes gráftranszformációs szabályok alkalmazásának időtartama veszi át.

A kutatás során a kidolgozott algoritmus is implementálásra került, amely első lépésben a Petri-háló optimalizálására alkalmazza a produkciós háló optimalizálására kidolgozott megoldástér szűkítő és optimalizálást gyorsító, majd a következő fázisban ellenőrzi az optimális megoldás tüzelhetőségét. A tüzelhető megoldások szűrésére a kutatási célra szabadon hozzáférhető BDD csomagokat és a SPIN modellellenőrző eszközt használtuk.

Ennek keretében kialakítottunk egy olyan környezetet, amely alkalmas a különböző megközelítésű modellezési paradigmák integrált hatékonyságvizsgálatára. A Petri-háló logikai struktúráinak vizsgálatát illetve a temporális kényszerek elemzését a Buddy nevű bináris döntési diagram alapú csomag illetve a SPIN nevű model checker segítségével végezzük. Az alárendelt mennyiségi optimalizálást részben az általános célú Ilog OPL Studio lineáris és vegyes egész optimalizálást támogató kereskedelmi csomaggal, részben pedig a Veszprémi Egyetemen kidolgozott PNS csomaggal végezzük. A modell transzformációs feladatokat részben a Berlini Műszaki Egyetemen kidolgozott AGG csomaggal részben pedig az BME MIT-n kidolgozott VIATRA2 általános transzformációs környezetben valósítottuk meg.

A kisminta kísérletek is azt mutatják, hogy műszaki szempontból is értelmes feladatokat lehet vizsgálni az új megközelítés segítségével. A vizsgálatok kifejező erejét jelentősen növeli az új eredmény, hogy a Petri-háló által adott strukturális és működési peremfeltételek kiegészíthetők a viselkedés megkívánt dinamikáját előíró temporális kényszerekkel. A korai hatékonyságvizsgálat azt mutatja, hogy a módszer hatékonyabban keres még a Veszprémi Egyetemen kidolgozott és nemzetközi élvonalbeli PNS technológiánál is az egész számok felett végzett keresések esetében.

A kutatás utolsó évében megvizsgáltuk, hogy miként lehet gyorsítani a kidolgozott algoritmust a logikai bázisok segítségével, melyeket a PNS technológiával adunk meg [36]. Ezen módszer lényege, hogy az egészek fölötti keresést redukáljuk azáltal, hogy nem a struktúra fölött optimalizálunk, hanem a struktúrálag lehetséges megoldási struktúrák alapján, vagyis a Branch-and-Bound algoritmusban nem az egyes műveleti

egységek bevétele alapján generáljuk a részfeladatokat, hanem a struktúrálag lehetséges bázis megoldási struktúrák alapján.

A kutatás utolsó szakaszában felmerült a kérdés, hogy miként lehet a probléma struktúra-alapú optimalizálását gyorsítani. Mivel a probléma lehetséges struktúrái logikai formulával megadhatók, ezért ezek halmaza kompakt formátumban, egy bináris döntési diagram formájában is ábrázolható [32]. A további kutatások célja, hogy összevessük az e bináris döntési diagram alapján végzett Branch-and-Bound algoritmus hatékonyságát a már kidolgozott módszerekével, valamint megvizsgálni, hogy e diagram milyen kapcsolatban áll a PNS technológiában használt „consistent decision mapping” módszer alkalmazásával.

## **Modellezés**

Tekintettel arra, hogy az alapul vett Petri-hálós és produkciós hálók szintézis módszerek egyaránt alacsony szintű modellekkel dolgoznak, e matematikai modelleknek az alkalmazási területek valamilyen magasszintű leírásából történő származtatására kellett megoldást adnunk. E problémára automatikus modelltranszformációs megoldást dolgoztunk ki.

Az automatikus modelltranszformációk kutatása területén a legfontosabb célkitűzés az volt, hogy olyan matematikailag is megalapozott, ugyanakkor hatékony módszereket dolgozzunk ki, amelyek lehetővé teszik a mérnöki gyakorlatban elterjedt modellező eszközök, elsődlegesen az UML (Unified Modeling Language) használatát a felhasználó számára, valamint a modelltől való bizonyítottan helyes módon történő matematikai modellek származtatását. E területen kiemelendő, hogy sikerült megoldani az informális mérnöki modellezés és a matematikai modellezés szabatos összekapcsolását [1,16]. E megközelítések nemzetközileg is újszerűnek minősülnek és pozitív visszhangot váltottak ki. Ezen általános keretbe foglalva, mintegy kiegészítő tevékenységként jelentek meg a szabályalapú transzformációhoz kapcsolódó mérnöki vizualizáció, programhelyességi bizonyítás kérdéseivel foglalkozó cikkek [4,7,16,17,18,20,33].

A kutatás során megvizsgáltuk, hogy hogyan lehet a magasszintű leírásokból elvégzendő transzformációkat matematikai tisztasággal definiálni. Kiemelt eredménynek tartjuk, hogy elsőként sikerült olyan metamodellezésen alapuló keretrendszert megfogalmazni, amely e szabatos megfogalmazáson túlmenően képes például olyan a modellalkotásban kulcsszerepet játszó műveletek precíz kezelésére, mint a modell absztrakció (e témában az UML közösség legrangosabb konferenciáján tartottunk előadást, amelyet folyóiratcikknek is meghívtak [5,7]).

Az alkalmazási területek előkészítésére vizsgálatokat végeztünk a termelési folyamatok modellezésére [8], valamint a diagnosztikai problémák modellezési lehetőségeinek vizsgálatára.

Ugyanakkor kísérleteket végeztünk az UML-nél egyszerűbb, de a gyakorlatban elterjedten használt olyan ipari workflow modellező eszköz illesztésére is, mint az IBM

Holosofx.

Komplex transzformációk esetén szükségessé válik az, hogy nem csak a modellek struktúráját kell módosítani, hanem az eredeti modellben felvett, becsült vagy mért jellemzőket is komplex matematikai transzformációknak kell alávetni. Ennek érdekében kiegészítettük az alapvető gráftranszformációs módszert a Berliini Műszaki Egyetemen létrejött együttműködés keretében [29,30]. Ez annyit jelent, hogy az általunk használt, alapvetően logikai megközelítésű gráftranszformációt kibővítjük a numerikus attribútumok kezelésével és erre az AGG-t, attribútumos gráfnyelvtanokat választottuk célmechanizmusnak. Ennek során kidolgoztunk egy a gráftranszformáció elvén működő automatikus modellgenerálási algoritmust, amely olyan UML modellekből származtatja az analízis alapját képező Petri-hálót, amelyeknél az erőforrásigény a szabványos General Resource Model profil segítségével adott.

UML oldalon ez a megközelítés nemcsak az informatikai megoldások erőforrás allokálásának szabványos megközelítése, hanem alkalmas arra is, hogy különböző munkafolyamatok erőforrás használatával bővített modelljéből optimalizálási modelleket lehessen származtatni (pl. egy jellegzetes alkalmazási terület a szoftver projektmenedzsment, melynél a fejlesztési folyamat modelljéhez a fejlesztői kapacitások allokálásának optimalizálása a célkitűzés).

### ***Diagnosztikai feladatok megoldása***

A kutatás során kidolgozásra került egy olyan általános algoritmus váza, amely alkalmas arra, hogy egy UML-ben leírt modellből automatikusan származtasson a diagnosztika alapjául szolgáló modelleket [10], valamint megvizsgáldott, hogy ezek milyen módon illeszthetők a műszaki gyakorlatban előforduló szabványokhoz [1]. A diagnosztikai metodika és az operációkutatás tartalmi összekapcsolásának előkísérletét a viszonylag legegyszerűbb struktúrákat nyújtó multiprocesszoros rendszerek körében végeztük el. [2,15]. Megfogalmazódott a multiprocesszoros valószínűségi diagnosztika P-gráfos formája és ezügyben futtatási kísérletek igazolták a módszer várható hatékonyságát [2].

Előkészületi munkálatok folytak a diagnosztika tesztalapú megközelítésére [9]. A távolabbi alkalmazásokat készítik elő azok a munkálatok, amelyekkel a diagnosztikai modelleknek kísérletes paraméterezését készítjük elő [14,34].

A kidolgozott metodika önmagában elégséges ahhoz, hogy diagnosztikai feladatokat UML alapon lehessen modellezni és azokból szabatosan matematikai optimalizálási feladatot generálni [37].

A módszerek alapjai beépültek a Typotex kiadó gondozásában megjelent „Formális módszerek az informatikában” című tankönyvbe [27], valamint nemzetközi disszeminációban [11-13].

## **Operációkutatás**

Operációkutatási oldalról a lineáris feltételrendszerrel adott, szeparábilis konkáv minimalizálási probléma területén értünk el eredményeket [26,35], mely problémára számos gyakorlati kérdés, mint bizonyos irányításelméleti feladatok, konkáv hátizsák probléma, termelési és szállítási feladatok, termelési folyamatok tervezése, folyamat hálózatszintézis feladatok, egyes hálózati folyam feladatok is vezetnek.

A cikkben megadjuk a lineáris korlátozási feltételekkel adott szeparábilis konkáv minimalizálási feladat egy elégséges optimalitási kritériumát. Az optimalitási kritérium, a feladat lineáris programozási relaxáltjának az érzékenység vizsgálatán alapul. A változók számának megfelelő számú tesztpont segítségével - sok esetben - eldönthetjük, hogy az adott bázis megoldásunk optimális-e vagy sem. Elégséges optimalitási kritériumunkat, korlátozás és szétválasztás típusú algoritmusban lehet megállási kritériumként felhasználni.