

## **Wprowadzenie**

Technologia satelitarna stała się podstawą dla funkcjonowania wielu pochodnych narzędzi wykorzystywanych w wielu dziedzinach współczesnego życia. Praca tych urządzeń w natychmiastowy sposób znalazła zastosowanie w nauce i sektorze militarnym. W późniejszym czasie umożliwiły prognozowanie zjawisk naturalnych, a przesyłanie sygnału telewizyjnego spowodowało wniknięcie tej technologii do życia codziennego ludzi i rozwój telewizji satelitarnej. Bardzo ważną rolę było zapewnienie łączności poprzez rozwój telekomunikacji oraz stworzenie systemów nawigacji satelitarnej, z której korzystają dzisiaj miliony ludzi na całym świecie. W życiu codziennym ludzie używają słowa „satelita” w znaczeniu potocznym, czyli w odniesieniu do satelity sztucznego stworzonego przez człowieka, który służy do przesyłania sygnału telewizyjnego lub nawigacyjnego GPS, robi zdjęcia Ziemi i umożliwia komunikację telefoniczną. W sensie ścisłym jest to generalnie dowolny obiekt poruszający się wokół innego obiektu w przestrzeni kosmicznej, po orbicie [2].

Temat ogólnie związany z technologią satelitarną jest bardzo szeroki. Dokonując przeglądu dostępnej literatury i materiałów z tego zakresu można szybko dostrzec, że dotyczy on szerokiego spektrum przeplatających się dziedzin nauki i techniki takich jak astrologia, różne płaszczyzny informatyki, elektronika, fizyka, wojskowość, wiele rodzajów nauk przyrodniczych, geodezja, chemia i wiele innych. Literatura wojskowa ujmuje przy tym temat z bardzo ciekawej perspektywy, ponadto jest pisana prostym i przystępnym językiem, choć w dość szczegółowy sposób. Studiując temat dotyczący technologii satelitarnych można bardzo często trafić na wiedzę, która dotyczy czynności i urządzeń z których korzysta się na co dzień, często jednak nie rozumiejąc na jakiej zasadzie one działają. Obecność satelitów (lub ogólnie obecność człowieka) w kosmosie stanowi relatywnie bardzo krótki okres. Zagadnienie to, wraz z eksploracją kosmosu jest i z pewnością w najbliższej przyszłości będzie jednym z najbardziej eksploatowanych dziedzin nauki i odkryć w różnych dziedzinach techniki i inżynierii.

Celem niniejszej pracy jest przedstawienie problemu harmonogramowania obserwacji satelitarnych wraz z informacjami potrzebnymi do kompleksowego zrozumienia tego zagadnienia, a także zaprezentowanie modelu DLP pozwalającego na rozwiązywanie tego problemu za pomocą narzędzi informatycznych. Ponadto intencją jest także dokonanie analizy zaproponowanego rozwiązania i badań polegających na porównaniu uzyskanych wyników.

## **Struktura pracy**

- wprowadzenie stanowi krótki i uogólniony wstęp do problematyki pracy, a ponadto określa jej podstawowe cele,

- w rozdziale pierwszym znajdują się informacje dotyczące organizacji operowania satelitów na orbitach, technologii satelitarnych i ich wykorzystania, które są konieczne do zrozumienia problematyki pracy. Przytoczona została krótka historia obserwacji satelitarnych i wiadomości dotyczące orbit wraz z klasyfikacją orbit. Rozdział zakończono zwięzłym nawiązaniem do teorii obliczeń, gdzie przytoczono podstawy dotyczące problemu harmonogramowania obserwacji satelitarnych z perspektywy informatyki oraz użyteczne objaśnienia w temacie programowania w logice,
- rozdział drugi to charakterystyka problemu oraz przegląd dotychczasowych badań, w których to podjęto już próbę rozwiązywania problemu harmonogramowania obserwacji satelitarnych. W tym rozdziale opisano nieco szerzej wybrane metody stosowane w przeszłości do rozwiązywania problemu,
- w rozdziale trzecim przedstawiono opis stworzonego na potrzeby badań modelu DLP. Rozdział przedstawia w pewnym sensie proces conceptualny w efekcie którego zaprojektowano i następnie zaimplementowano model rozwiązania problemu harmonogramowania (w tym szczegółowe założenia dotyczące formalnej definicji problemu),
- rozdział czwarty to przedstawienie wyników oraz szczegółowa analiza przeprowadzonych badań z wykorzystaniem stworzonych na te potrzeby narzędzi, w tym głównie modelu DLP. Analiza poprzedzona jest objaśnieniem przyjętej metodologii badań wraz z naświetleniem niezbędnych definicji.

# 1. Informacje ogólne

W tym rozdziale umieszczono ogólne ramy teoretyczne potrzebne do konceptualnego zrozumienia problemu harmonogramowania obserwacji satelitarnych. Zawarto w nim wybrane elementy historii obecności sztucznych satelitów w przestrzeni kosmicznej i podstawowe informacje dotyczące operujących na różnych orbitach Ziemi satelitów. Opisano także najważniejsze przykłady technologii bazujących na danych i komunikacji z satelitami oraz przykłady praktycznego wykorzystania danych z obserwacji satelitarnych Ziemi. Przedstawiono także informacje ściśle związane z technologiami i narzędziami informatycznymi wykorzystanymi na potrzeby badań wykonanych w zakresie harmonogramowania obserwacji satelitarnych będących celem niniejszej pracy.

W pracy przyjęto następujące definicje kluczowych pojęć:

- harmonogramowanie – planowanie prac poprzez przydzielanie zasobów pracy do zadań oraz ustalenie rozkładu wykonywania tych zadań w czasie [1],
- satelita obserwacyjny – urządzenie umieszczone na orbicie Ziemi, którego zadaniem jest detekcja i pozyskanie danych dotyczących określonych obszarów na powierzchni planety zgodnie z potrzebami, a także dostarczenie tych danych do stacji naziemnej [3],
- nadir - punkt przecięcia powierzchni Ziemi z odcinkiem łączącym środek planety i satelity [13].
- zadanie obserwacyjne – czynność wykonywana przez satelitę obserwacyjnego znajdującego się na orbicie polegająca na zgromadzeniu wymaganych danych dotyczących obszaru na powierzchni planety. Każde zadanie ma określone koszty jego wykonania,
- awaryjne zadanie obserwacyjne – szczególny typ zadania obserwacyjnego, wymagający szybkiego i obligatoryjnego wykonania,
- priorytet zadania obserwacyjnego – wartość liczbową wyrażającą korzyść z wykonania danego zadania obserwacyjnego. Im wyższy priorytet, tym wyższa korzyść z wykonania zadania,
- okno czasowe widoczności – przedział czasu, w którym możliwe jest wykonanie danego zadania,
- okno czasowe zadania – przedział czasu w obrębie okna czasowego widoczności, w którym dane zadanie jest wykonywane,
- stacja TTC&M (ang. tracking, telemetry, command and monitoring) – [4] umożliwia wszelką komunikację z satelitą w celu wymiany danych, jej zarządzania, kontroli i sterowania,
- stacja kontrolna - rodzaj stacji TTC&M, którego zadaniem jest przysyłanie danych dotyczących zarządzania satelitami,
- stacja naziemna - rodzaj stacji TTC&M, którego zadaniem jest odbiór danych zgromadzonych w pamięci komputera satelitów,

- model DLP – (model rozwiązywania problemu) – uogólniony, deklaracyjny opis rozwiązania danego problemu, zapisany w formie faktów i reguł, niezależny od konkretnej postaci (instancji) problemu,
- instancja problemu – konkretny, rzeczywisty przypadek danego problemu pozwalający na analizę w celu znalezienia rozwiązania.

## 1.1 Historia obserwacji satelitarnych

Historia obserwacji satelitarnych, a w zasadzie obecności stworzonych przez człowieka satelitów na orbicie Ziemi zaczęła się w roku 1957 w ówczesnym Związku Radzieckim, gdzie dokonano udanego wystrzału satelity noszącego nazwę SPUTNIK 1 na orbitę Ziemi. Były to czasy tzw. wyścigu o kosmos, - rywalizacji USA i ZSRR na płaszczyźnie militarno-gospodarczo-naukowej w podboju przestrzeni kosmicznej będącą elementem zimnej wojny, którego elementem był pierwszy sztuczny satelita [13]. Kolejnym ważnym punktem w historii było utworzenie rok później w USA agencji kosmicznej NASA (National Aeronautics and Space Administration), dysponującej zasobami umożliwiającymi prowadzenie działań w obszarze lotów kosmicznych [13]. W tym samym roku wystrzelono w Stanach Zjednoczonych pierwszego na świecie satelitę mającego służyć celom badawczym EXPLORER 1. Seria explorer liczyła kilkadziesiąt satelitów, a popularnym osiągnięciem właśnie pierwszego z tej serii było odkrycie pasów promieniowania Van Allena [13]. Późniejsze próby uruchamiania coraz bardziej zaawansowanych satelitów sztucznych na orbicie planety odzwierciedlają potrzeby ludzkości, na zaspokojenie których odpowiedziała inżynieria i technologia kosmiczna, gdzie coraz większą rolę odgrywały komputery o coraz większej mocy obliczeniowej i efektywności. Przez lata zastosowanie łączności satelitarnej było jedynym sposobem zapewnienia komunikacji między kontynentami Ziemi. Operujące satelity zapewniające komunikacje to na przykład: Telstar, Syncom, Molniya, czy seria INTELSAT. Umieszczenie kabli podmorskich o dużej przepustowości przejęło jednak to zadanie w ostatnich latach [9]. Początki telewizji satelitarnej w drugiej połowie XX w. były związane z uruchomieniem kolejnych urządzeń: satelity Westal i satelitów serii Eutelsat Hot Bird, które operują do dziś. Następnym ważnym momentem w historii technologii kosmicznej był start satelity **TIROS 1** (Television Infrared Observational Satellite), był to satelita meteorologiczny rozpoczynający serię urządzeń gromadzących dane i wykonujących fotografie naturalnych zjawisk atmosferycznych występujących na powierzchni Ziemi, które można było odtąd zobaczyć z zewnątrz planety (z kosmosu) [13]. W późniejszym czasie doszło do startu ulepszonych satelitów pogodowych dysponujących coraz nowszymi technologiami i sensorami (np. program Nimbus).

Oczywistą możliwością wykorzystania satelitów obserwacyjnych stało się wtedy poszukiwanie złóż surowców zarówno na lądzie, jak i pod powierzchnią mórz i oceanów. Kluczowymi satelitami służącymi przy wykonywaniu zadań obserwacyjnych Ziemi min. w tym zakresie do dziś są satelity

serii „Landsat”, która to rozpoczęła się w roku 1972 uruchomieniem pierwszego satelity, pierwotnie pod nazwą „Earth Resources Technology Satellite 1” (w 1975 r. zmieniono na Landsat 1, która wymownie podkreśla ogólne wykorzystanie w zakresie zdalnego obserwowania powierzchni Ziemi, niekoniecznie związane z poszukiwaniem i badaniem zasobów naturalnych) [10]. Landsat dysponował nowoczesnym skanerem multispektralnym zapewniającym bardzo dobrą jakość zdjęć z obserwacji [13]. Seria Landsat trwa do dziś, a dane i fotografie wykonane przez te satelity są ogólnodostępne. W związku z sukcesem serii Landsat uruchomiono roku 1978 serię Seasat urządzeniem SEASAT 1 dysponującym aparaturą wykorzystującą mikrofałę do stosunkowo dokładnego pomiaru temperatury, kierunku i prędkości wiatru oraz obrazowania układu ziemskiej pokrywy lodowej, prądów morskich i chmur [13].

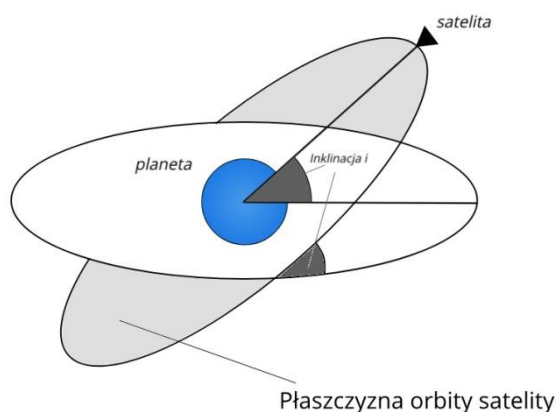
W późniejszym czasie zaczęto konstruować inne urządzenia umożliwiające różne metody eksploracji przestrzeni kosmicznej z orbit naszej planety. W tym obszarze wielkim sukcesem okazał się teleskop kosmiczny Hubble, będącego efektem współpracy NASA i ESA. Wyniesiony przez prom kosmiczny rozpoczął pracę w roku 1990. Również kolejne teleskopy kosmiczne zaczęły funkcjonować w późniejszych latach 90-tych i na początku XXI wieku. Teleskopy kosmiczne umożliwiły wielokrotnie lepszą jakość zdjęć przestrzeni kosmicznej i zjawisk w niej panujących niż byłoby to możliwe z powierzchni Ziemi, z której obserwacje odległych o miliardy lat świetlnych obiektów byłyby po prostu nieosiągalne [13].

Jednym z ważniejszych osiągnięć ogólnie w obszarze eksploracji kosmosu było także skonstruowanie Międzynarodowej Stacji Kosmicznej ISS (International Space Station) przy wspólnym udziale 17 krajów (Stanów Zjednoczonych, Kanady, Rosji, Japonii, Brazylii i 11 państw budujących i wspierających Europejską Agencję Kosmiczną). Wyniesiona na orbitę o wysokości ponad 400 km i inklinacji 51,6 stopni pozwala na efektywne badania różnego rodzaju zjawisk naturalnych w warunkach bardzo niskiej grawitacji, ma duży potencjał obserwacyjny i umożliwia stosunkowo łatwe utrzymanie we współpracy naukowców z różnych miejsc na Ziemi [13]. Obecnie szeroko rozumiane misje kosmiczne obejmują coraz większy zakres działań badawczych i sięgają w coraz bardziej oddalone od Ziemi miejsca w przestrzeni kosmicznej.

## **1.2 Elementy i rodzaje orbit**

**Orbita** wg [2] to „droga obiektu krążącego wokół przyciągającej go masy w przestrzeni, na przykład planety krążącej wokół Słońca lub satelity wokół planety”. Orbity wykorzystywane przez satelity stworzone przez człowieka są także zwane orbitami keplerowskimi (odniesienie do Jana Keplera), a określenie to dotyczy oddziaływania na siebie dwóch mas punktowych, do których dochodzi w efekcie przyciągania grawitacyjnego [9].

**Inklinacja orbity** to nachylenie jej płaszczyzny w stosunku do badanej płaszczyzny masy wokół której orbituje. W przypadku satelity Ziemi kąt ten mierzony jest do płaszczyzny równika. Satelita o przykładowej inklinacji wynoszącej 63 stopnie, będzie przemieszczał się (relatywnie do Ziemi) w kierunku północnym, ostatecznie przemieszczając się nad punktem o szerokości geograficznej 63 stopni N. Następnie ten relatywny ruch będzie odbywał się w kierunku południowym przemieszczając się nad punktem 63 stopni S. Inklinalacja orbity operującego satelity w połączeniu z ruchem obrotowym Ziemi wokół własnej osi ma więc bezpośredni wpływ na rozkład punktów na Ziemi nad którymi przemieszcza się satelita [13].



**Rysunek 1. Inklinalacja orbity**

Orbit

y można sklasyfikować na wiele sposobów. W zależności od przeznaczenia i wymagań dotyczących danego satelity jest on umieszczany na orbicie pozwalającej na wykonywanie przez niego oczekiwanych zadań (np. obserwacyjnych). Im mniejsza wysokość orbity i przemieszczającego się po niej satelity, tym lepszą rozdzielczość obrazu można uzyskać, co wprost wiąże się z jakością dostarczanych danych. Niska orbita będzie oznaczać także krótszy czas obiegu wokół planety, a więc większą częstotliwość jeśli chodzi o widoczność tego samego obszaru na powierzchni planety. Ponadto wiąże się to również z czasem potrzebnym na transfer danych pomiędzy satelitą a stacją na powierzchni planety. Im niżej znajduje się satelita, tym bliżej jest on powierzchni planety, a więc mniejsza jest droga, którą musi przebyć impuls elektromagnetyczny pomiędzy satelitą, a komunikującą się z nim stacją. Kolejną rzeczą jest pole obserwowalnej powierzchni planety. Obserwowalny obszar jest tym wyższy, im bardziej urządzenie oddalone jest od powierzchni Ziemi [13]. Bazując na powyższych zasadach, przyjmując za kryterium wysokość orbity na której operuje satelita można wyróżnić 4 podstawowe typy orbit:

### **1. Orbita niska (LEO – Low Earth Orbit)**

Satelita krążący na orbicie niskiej znajduje się na wysokości 160 km – 1600 km. Taka odległość zapewnia wysoką rozdzielczość i łatwą komunikację ze stacją. Warto odnotować, iż wprowadzenie satelity na orbitę niską jest także technicznie łatwiejsze (np. rakieta wymaga mniej paliwa), jednak

trzeba mieć na uwadze, że atmosfera znajdująca się na tych wysokościach może mieć negatywny wpływ na pracę operującego satelity. Okres obiegu satelity wprowadzonego na taką orbitę to około 100 minut [13]. Szczególnym, ważnym przypadkiem niskiej orbity jest orbita heliosynchroniczna o odpowiedniej wysokości, której zasadniczą zaletą, jest możliwość obserwacji dokładnie tego samego punktu na powierzchni planety o dokładnie tej samej porze dnia (ten sam czas słoneczny). Identyczne naświetlenie obszaru przez światło słoneczne pozwala na efektywne obserwacje terenu i zjawisk meteorologicznych. Poza tym satelity funkcjonujące na orbitach niskich służą głównie monitorowaniu zjawisk pogodowych, czy ogólnym obserwacjom dla celów naukowych[13].

## **2. Orbita średnia (MEO – Medium Eart Orbit)**

Orbity średnie charakteryzują się wysokością w zakresie 1600 km – 19300 km, a okres obiegu satelity wprowadzonego na taką orbitę zmienia się w zależności od wysokości w zakresie od 100 minut do 12 godzin [13]. Większa odległość satelity na orbicie średniej od obserwowanego obszaru zapewnia dłuższe okno czasowe widoczności tego obszaru. Korzyścią jest tutaj również pomijalny wpływ atmosfery. Satelity operujące na orbitach średnich dostarczają danych głównie dla systemów nawigacyjnych i telekomunikacyjnych.

## **3. Orbita geosynchroniczna (GEO – Geosynchronous Eart Orbit)**

Przypadek orbity geosynchronicznej dla urządzenia satelitarne zachodzi, gdy zostanie ono umieszczone na wysokości  $\sim 35\,786$  km. Orbitujący satelita porusza się wtedy synchronicznie względem obrotu kuli ziemskiej (okres obiegu satelity jest wtedy równy 1 dobie ziemskiej), czego efektem jest to, że pozostaje on praktycznie stale nakierowany na jeden punkt na powierzchni planety, poruszając się kątowno równo z nią. Obserwowalny obszar dla satelity umieszczonego na orbicie geosynchronicznej obejmuje dużą część powierzchni całej planety, co umożliwia jego efektywne wykorzystanie w komunikacji, obserwowaniu zjawisk pogodowych i szybkim informowaniu o nadchodzących zagrożeniach [13]. Szczególnym przykładem orbity geosynchronicznej jest orbita geostacjonarna. Ten przypadek zachodzi, kiedy satelita orbituje dokładnie po linii równika (orbita o inklinacji 0). Orbita geostacjonarna zawsze jest orbitą geosynchroniczną, jednak orbita geosynchroniczna o innej inklinacji (szerokości geograficznej) niż 0 nie będzie geostacjonarną [13]. Warto odnotować, że trudnością dotyczącą stosowania orbity geosynchronicznej, z perspektywy niektórych państw jest samo umieszczenie na niej satelity, gdyż najlepsze warunki do startu panują na równiku.

## **4. Orbita silnie eliptyczna (HEO - Highly Elliptical Orbit)**

Ten typ orbity różni się od poprzednich sposobem poruszania się wokół planety. Poprzednie orbity były kołowe (droga krążącego orbity tworzy okrąg), w tym przypadku jest to elipsa. Mimośrodowy ruch powoduje, że wysokość satelity zmienia się, cyklicznie zbliżając się i oddalając od powierzchni planety (jest to około 1000 km – 38 000 km). Zmianie ulega również prędkość

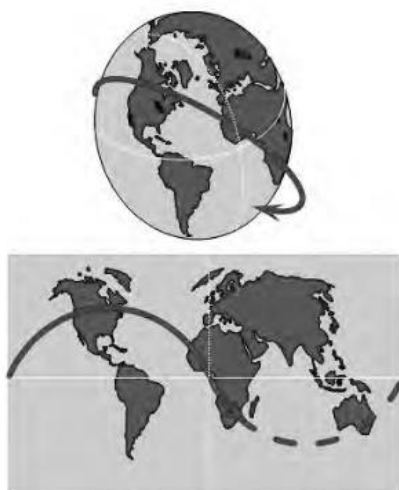
poruszania się satelity w przestrzeni kosmicznej – im dalej planety, tym mniejsza. W związku z tym przedział czasu w którym dany obszar jest obserwowalny może być bardzo długi, właśnie ze względu na długi czas przebywania satelity w apogeum orbity (punkcie najbardziej oddalonym). Satelity operujące na orbitach silnie eliptycznych mają zastosowanie na przykład w telekomunikacji lub celach militarnych i wywiadowczych.

### **Naziemny tor ruchu satelity**

Relatywny ślad ruchu satelity nad powierzchnią Ziemi zależy od rodzaju obranej orbity – wysokości, inklinacji itd. Parametry te wyznaczają punkty na powierzchni planety, bezpośrednio nad którymi przemieszcza się satelita (nadir). Linia, którą tworzą te punkty tworzy naziemny tor satelity, który można zaznaczyć na globusie. Przebieg linii toru naziemnego satelity jest bardzo ważny w planowaniu i zarządzaniu operującymi satelitami, bowiem stacjonujące na powierzchni Ziemi stacje zarządzające muszą być w zasięgu satelitów, aby sprawnie się z nimi komunikować.

### **Koszty startu satelity**

Start satelity to bardzo kosztowne przedsięwzięcie. Średni koszt związany z samym umieszczeniem satelity na orbicie zależy głównie od masy urządzenia i wysokości orbity, na której docelowo ma operować satelita. Jest to związane z ilością paliwa potrzebną przy starcie rakiety, a przykładowy koszt startu to \$180 mln. (pojazd Ariane 5, orbita geostacjonarna - Europejska Agencja Kosmiczna, 2000 r.) [13].



***Rysunek 2. Naziemny tor ruchu satelity [13]***

## **1.3 Obserwacje satelitarne we współczesnym świecie**

Określenie sumy działających współcześnie urządzeń satelitarnych służących do celów obserwacyjnych lub ogólnie orbitujących Ziemię może okazać się trudne. Według organizacji Union of Concerned Scientists w kwietniu roku 2020 na orbitach naszej planety znajdowało się w



sumie 2667 satelitów, część może być nieaktywna lub pozostawiona do aktywowania w razie nagłego wypadku. Szczegółowe informacje w tym zakresie być może są niedostępne lub nieaktualne. Według tej samej bazy danych w grupie wszystkich orbitujących satelitów łącznie 885 to urządzenia obserwacyjne, w których są min. charakterystyczne serie urządzeń takie jak Landsat, Dove, Lemur itd. [a]. Z dostępnych danych wynika również, że zdecydowana większość satelitów umieszczana jest na orbitach niskich, być może ze względu na mniejsze koszty startu w stosunku do innych orbit oraz możliwości, które sama w sobie stwarza pozycja satelity na tej orbicie. Wybór orbity LEO stwarza również możliwość orbitowania heliosynchronicznego (pojęcie to zostało wyjaśnione wcześniej), które jest szczególnie korzystne w przypadku wykonywania zdjęć powierzchni planety [a]. Ponadto, przyjmując jako kryterium dotychczasowy dorobek poszczególnych państw na płaszczyźnie technologii satelitarnej, w ilości udanych startów pozwalających na uruchomienie satelity dominują Stany Zjednoczone, Chiny i Rosja [a].

### **Współczesne zastosowania technologii satelitarnych**

Główne systemy bezpośrednio oparte na danych satelitarnych z urządzeń krążących wokół Ziemi to systemy nawigacji satelitarnej, wszelkie systemy teledetekcji i obrazowania powierzchni planety oraz sieci łączności globalnej. Współcześnie bardzo popularne są przede wszystkim systemy nawigacji oraz systemy teledetekcji, których spektrum zastosowania jest ogromne.

### **Nawigacja satelitarna**

Systemy nawigacji satelitarnej (GNSS – Global Navigation Satellite Systems) to satelitarne systemy określania pozycji i prędkości obiektu oraz czasu, powszechnie dostępne w sposób ciągły dla ludzi [8]. Systemy te można podzielić na globalne i regionalne. Systemy globalne umożliwiają nawigację na powierzchni całej planety. Te ostatnie służą jako systemy w mniejszych państwach, które wspomagają systemy globalne oraz umożliwiają większą precyzję i dostępność w kontekście obszarów lokalnych [8]. We współczesnym świecie dostępne są następujące globalne systemy nawigacji satelitarnej

- GPS (NAVSTAR) – system amerykański,
- GLONASS – system rosyjski,
- Galileusz – system europejski,
- Compass – system chiński,

Większość z nich jest w dużym stopniu pod kontrolą wojskową, wyjątkiem jest europejski „Galileusz”, który jest całkowicie pod kontrolą organizacji cywilnych [8].

### **Łączność satelitarna**

Współczesne życie w bardzo dużym stopniu opiera się na szybkiej wymianie informacji, w dzisiejszym świecie ludzie na całym świecie korzystają z sieci telefonii komórkowej. Oprócz technologii umożliwiającej komunikację za pomocą naziemnych stacji bazowych dostępna jest

także łączność satelitarna. Do systemów satelitarnych należą min. Iridium i Globalstar. Łączność satelitarną uzyskuje się przy pomocy szeregu satelitów umieszczonych na niskich orbitach, których zadaniem jest zapewnienie łączności pomiędzy punktami komunikującymi się między sobą na powierzchni Ziemi. Wykorzystanie orbit niskich skraca drogę impulsów elektromagnetycznych (mniejsze opóźnienie) i jednocześnie koszt energetyczny sygnału. Do infrastruktury tych systemów należą dodatkowo bramki naziemne (w przypadku Iridium) lub sieć naziemnych stacji pomocniczych (Globalstar) realizujących połączenie. Odbiorcami tych systemów są głównie strategiczne sektory państw lub branże wykonujące prace w trudnym terenie [8].

### **Obserwacje satelitarne**

Satelity obserwacyjne dostarczają zdjęć terenu i informacji meteorologicznych, których nie można pozyskać z powierzchni Ziemi. Są to złożone instrumenty badawcze, pozyskujące dane w wielu kanałach spektralnych, stąd są nazywane aparatami hiperspektralnymi. Oznacza to, że dany satelita obserwacyjny dysponuje szeregiem sensorów rejestrujących pewien typ fal elektromagnetycznych (o pewnym przedziale długości fali). Co więcej - zbadano, że każdy z tych typów fal jest skorelowany w pewien sposób z różnymi procesami lub wielkościami fizykochemicznymi takimi jak: wilgotność, temperatura, ilość chlorofilu w roślinach, charakter soli morskiej, czy czystość wód. Wyznaczono tzw. wskaźniki różnicowe wyznaczalne na podstawie danych satelitarnych poszczególnych kanałów spektralnych satelity, które dobrze wizualizują pewne zjawiska - podkreślają je przy jednoczesnym stłumieniu innych (efektem jest uwypuklenie obszarów na których panują badane warunki fizyczne, chemiczne, biologiczne itd.) [12]. Charakterystycznymi systemami obserwacyjnymi są min. System Obserwacji Ziemi EOS (Earth Observing System) z satelitami **Terra** i **Aqua** zarządzany przez NASA oraz europejski METEOSAT. Do zarejestrowania danych satelita używa całego zestawu czujników światła, ciepła i promieniowania. Urządzenia te zwykle umieszczane są na orbitach geostacjonarnych lub orbitach polarnych o inklinacji ok. 90 stopni. Operujący w systemie satelita wykonuje obserwacje pokrywy chmur, pomiary temperatury, nawodnienia terenu (lub różnic w nawodnieniu), charakterystyki atmosfery, promieniowania oraz wzajemnej wymiany energii i pierwiastków naturalnych w przyrodzie np. cykl obiegu wody. Badanie tych zjawisk na płaszczyźnie globalnej ma dostarczać przede wszystkim informacji potrzebnych do zrozumienia funkcjonowania i wpływu elementów obserwowanych zjawisk na klimat planety [13]. Dane gromadzone w trakcie obserwacji satelitarnych w dużej części są dostępne dla przeciętnego człowieka z dostępem do sieci internet. Poprzez strony internetowe Europejskiego Programu Obserwacji Ziemi „Copernicus” umożliwiony jest dostęp do danych z satelitów z serii Sentinel. Ponadto podstawowym źródłem danych (a także innych informacji z różnych dziedzin nauk o Ziemi) są domeny United States Geological Survey (agencji naukowo-badawczej Stanów Zjednoczonych), agregujące dane dotyczące obserwacji satelitarnych

wykonanych przez urządzenia serii Landsat. <https://earthexplorer.usgs.gov/>. Na bazie tych danych możliwe było utworzenie komercyjnych aplikacji pobierających, analizujących i obrazujących dane satelitarne, co pozwala na wyciąganie wniosków w biznesie i zarządzaniu, przemyśle, rolnictwie [12].

Monitoring środowiska roślinnego – analiza i obrazowanie zestawu danych satelitarnych z obserwacji upraw roślin i obszarów wegetacji pozwala na wyciąganie wniosków dotyczących optymalnego poziomu nawożenia i nawodnienia, chorób zagrażających uprawom itp. Określono wskaźnik NDVI (znormalizowany różnicowy wskaźnik wegetacji), który bazuje na danych zgromadzonych przez sensory o kilku różnych kanałach spektralnych jest w stanie zobrazować stan flory na danym obszarze. Dzięki temu możliwe jest określenie obszarów o dobrze nawodnionej i gęsto rozwiniętej roślinności w kontraście do obszarów miejskich i zbiorników wód. Wykorzystanie obserwacji satelitarnych z analizą pochodzących z tych obserwacji danych jest bardzo popularne w rolnictwie, ekologii i klimatologii [12]. Inny ze wskaźników różnicowych – NDWI (znormalizowany różnicowy wskaźnik wody) dotyczy analizy stanu zbiorników i cieków wodnych, a w szczególności czystości lub zanieczyszczenia wody. Pozwala to w bezpośredni sposób zidentyfikować wody czyste oraz przeciwnie – źródła z których mogą pochodzić zanieczyszczenia środowiska (np. przemysłowe) [12].

W dziedzinie zarządzania kryzysowego możliwa jest szybka detekcja pożarów, identyfikowania obszarów zagrożonych występowaniem pożarów, osuwisk lub powodzi. W biznesie stosuje się połączenie wielu rodzajów danych z wielu dziedzin w celu wspomagania podejmowania decyzji o charakterze gospodarczym i finansowym oraz w celu estymowania ryzyka. Dotyczy to rynku ubezpieczeń, transportu, turystyki, górnictwa. Analiza różnych danych w połączeniu z danymi satelitarnymi w celu przewidywania cen różnych produktów na rynku, np. ropy lub produktów spożywczych, których ceny związane są z warunkami pogodowymi w danym roku. Ponadto dostępne są również aplikacje i agregaty danych służące masowemu odbiorcy przedstawiające podstawowe dane i fotografie - Google Maps, aplikacje prognozujące i obrazujące pogodę, aplikacje turystyczne, aplikacje prezentujące różnego rodzaju zanieczyszczenia gleb i wód, a także zanieczyszczenia powietrza w różnej formie (pyły lub toksyny), które obrazują zanieczyszczenie na pewnym większym obszarze, w przypadku stacji pomiarowych, badanie jest mniej dokładne (pomiar wykonywany jest punktowo a wyniki interpolowane na obszar) [13]. Z fotografii satelitarnych i innych danych (wykonanych na przykład przy pomocy wiązki lasera i sensora umieszczonego na satelicie) z powodzeniem korzysta branża budowlana w celu wykonania badań gruntu pod konstrukcję budowli lub dokładne określenie projektu i charakterystycznych punktów konstrukcji.

## 1.4 Przyszłość technologii satelitarnej

<#>Na koniec – przytoczyć KRÓTKO zagadnienia</#>

- miniaturyzacja satelitów obserwacyjnych
- CubeSat
- przepełnienie orbit
- kosmiczne śmieci
- krótko o możliwościach uczenia maszynowego przy analizie danych z obserwacji satelitarnych

## 1.5 Złożoność problemów obliczeniowych

Problem, który można przedstawić do rozwiązania w formie zrozumiałej dla maszyny liczącej (np. komputera) zwany jest problemem obliczeniowym. Problem obliczeniowy charakteryzuje się danymi wejściowymi o mniejszym lub większym rozmiarze oraz zdefiniowanym, przyjmowalnym rozwiązaniem tego problemu. Problemy obliczeniowe występują w różnych rodzajach. Rozwiązanie niektórych z nich jest łatwe i szybkie, inne zaś wymagają dużej ilości zasobów obliczeniowych (mocy obliczeniowej, pamięci). Na przykład: mając listę elementów danego typu (np. liczb), znalezienie elementu o największej (lub najmniejszej) wartości jest problemem łatwym do rozwiązania. Wymaga to bowiem jedynie przeanalizowania kolejnych elementów tej listy porównując jednocześnie z największą znaną dotąd wartością. Innym przykładem problemu obliczeniowego może być problem wędrownego sprzedawcy, w którym występuje pewna liczba miast i określona odległość między tymi miastami. Problem polega na odnalezieniu najkrótszej drogi przy której sprzedawca rozpoczynając i kończąc wędrowkę w jednym z miast odwiedzi pozostałe z nich w czasie podróży. Problem opisany w ten sposób wymaga przeanalizowania wszystkich kombinacji możliwych do obrania ścieżek, co stanowi dużą trudność, ponieważ już przy niewielkiej liczbie miast liczba tych kombinacji jest bardzo duża. Ostatecznie algorytm podejmujący próbę rozwiązania tego problemu, pracujący przy danych o pewnym, relatywnie niedużym rozmiarze nie znajdzie rozwiązania mając do dyspozycji czas, który byłby akceptowalny w ludzkim postrzeganiu. Mowa tu o tzw. warunku efektywności algorytmu, który – w tym przypadku nie będzie spełniony. W związku z tym, można opisać problem w nieco inny sposób, tj. założyć pewną oczekiwaną długość drogi  $n$  i spróbować znaleźć odpowiedź na to, czy istnieje trasa o łącznej długości mniejszej lub równej  $n$ . Jest to tzw. wersja decyzyjna tego problemu, która charakteryzuje się inną złożonością. Ten problem jest łatwiejszy do rozwiązania, zatem na tym przykładzie można dostrzec, że zmodyfikowanie danego problemu może umożliwić jego efektywne rozwiązanie, które może okazać się dość dobrym, aby wyciągnąć konkretne wnioski, albo wykorzystać je w praktyce [6].

W rzeczywistości występują także problemy, przy których rozważana jest nie złożoność, a obliczalność problemu, czyli możliwość znalezienia rozwiązania. Istnieją bowiem problemy, których nie można rozwiązać dotąd na żadnej maszynie liczącej [6].

Na złożoność danego algorytmu składa się czasowa złożoność obliczeniowa i pamięciowa złożoność obliczeniowa.

Klasy złożoności obliczeniowej

- P

- NP

<#>Króciutko, pobieżnie złożoność obliczeniowa</#>

## 1.6 Dysjunkcyjne programowanie logiczne

Dysjunkcja w przypadku programowania logicznego to konstrukcja polegająca na niewspółzachodzeniu – czyli eksklusywnej alternatywie (exclusive OR), czyli **a ALBO b**. Wg [1] dysjunkcja to: „w logice: zdanie złożone z dwóch zdań połączonych spójnikiem "albo", prawdziwe pod warunkiem, że zachodzi najwyżej jedna z dwóch wskazanych w nich możliwości; zaprzeczenie koniunkcji; niewspółzachodzenie;” W zrozumieniu pomaga zestawienie na zasadzie analogii do definicji koniunkcji: wg [1] koniunkcja to: „w logice: zdanie złożone ze zdań połączonych spójnikiem "i", prawdziwe wtedy, gdy każde ze zdań składowych jest prawdziwe;”. Tablica prawdy wizualizuje działanie dysjunkcji przy dwóch wyrazach logicznych a, b dla różnych wartości binarnych:

a	b	A XOR b
0	0	0
0	1	1
1	0	1
1	1	0

Programowanie logiczne (programowanie w logice) jest paradygmatem programowania opartym wprost na logice i wnioskowaniu. W przypadku popularnego współcześnie podejścia obiektowego (będącego w istocie paradygmatem imperatywnym) rozwiązanie problemu otrzymuje się przez modelowanie rzeczywistości za pomocą struktur danych zamkniętych w modułach, po czym należy stworzyć formalną listę kroków, które musi wykonać procesor komputera wykorzystując posiadane w pamięci dane w celu wygenerowania rozwiązania problemu. Jeśli chodzi o programowanie w logice modelowanie odbywa się przez zapisanie problemu w postaci czysto logicznej – jako zbiór twierdzeń, związków między twierdzeniami i sposoby wnioskowania jednych twierdzeń z drugih. Zapis ten to fakty i reguły które dotyczą badanego problemu. Kluczowe w programowaniu logicznym jest wnioskowanie (dowodzenie twierdzeń), czyli generowanie nowych faktów

opisujących rzeczywistość [14]. „Zaletami programowania w języku logiki jest to, że programy są deklaratywne i jednocześnie proceduralne (strukturalne). Wiemy, co program wylicza, zaś mniej już nas interesuje, jak to jest wyliczane” [cyt. 14].

## 1.7 Answer Set Programming

Answer Set Programming (ASP) jest techniką programowania w którym rozwiązanie problemu polega na wyszukaniu zbioru rozwiązań (answer set) w postaci tzw. stabilnych modeli z przestrzeni wszystkich możliwych odpowiedzi na zadany problem. Jest zaprojektowany do efektywnego wnioskowania i reprezentacji wiedzy opartym na paradygmacie programowania logicznego. Jest sposobem deklaratywnego modelowania wiedzy, pozwalający na rozszerzanie jej bazy, czyli wnioskowanie faktów na podstawie reguł. Wcześniej wykorzystywano jego możliwości przy badaniach i implementacji systemów sztucznej inteligencji oraz innych systemów silnie związanych z zarządzaniem wiedzą. ASP stosowane było głównie przez środowisko naukowe, w obecnych czasach trwają coraz bardziej rozpowszechnione prace nad zastosowaniem jego możliwości w przemyśle. ASP jako forma programowania pozwala na rozwiązywanie problemów obliczeniowych. Polega to na zdefiniowaniu problemu w postaci programu logicznego w taki sposób, aby wywnioskowana baza wiedzy reprezentowała związane z problemem odpowiedzi na zadany problem, a następnie na wykorzystaniu tzw. answer set solvera do wyszukania poprawnych rozwiązań zwanych zbiorem stabilnych modeli (answer set - opis akceptowalnego rozwiązania problemu informacjami w formie faktów) [24].

Obecnie dostępna jest grupa narzędzi ułatwiająca pracę z ASP pod nazwą “Potsdam Answer Set Solving Collection” (Potassco). Narzędzia te zostały stworzone w C++ i opublikowane pod licencją GNU na popularne platformy. Obejmują między innymi tzw. answer set solver (ang. solving – rozwiązywanie) „clasp” oraz tzw. grounder (ang. grounding – uziemianie, zakotwiczenie) „gringo”, które grają główną rolę podczas pracy z systemem ASP. Narzędzie „gringo” zwane grounderem ma za zadanie dokonanie analizy programu logicznego, napisane w języku gringo (zwykle zawierającego zmienne) i jego translacja na semantycznie identyczny program logiczny, nie zawierający już zmiennych (tzw. kotwiczenie, grounding). W następnym kroku może on być łatwo przetwarzany przez wybrany solver. Solver „clasp” przetwarza wynik działania groundera w celu obliczenia z wejściowego programu logicznego zbioru stabilnych modeli (answer sets) [5].

**clingo** to zintegrowany system ASP służący do rozwiązywania problemów łącząc rolę generowania możliwych modeli rozwiązań za pomocą wnioskowania z rozwiązywaniem przedstawionego problemu przy użyciu answer set solvera. Jego nazwa pochodzi od połączenia nazw dwóch narzędzi, wchodzących w jego skład – solvera „clasp” oraz groundera „gringo”. Celem zastosowania clingo jest ułatwienie pracy z ASP. W związku z tym system „clingo” zarządza

dwiema czynnościami, tj. groundingiem (gringo) oraz zwracaniem końcowych wyników w postaci zbioru stabilnych modeli (clasp) [5]. Podczas pracy z systemem ASP „clingo” korzystamy języka modelowania (programowania) dla „gringo”, jego gramatyka określa dozwolone ramy tworzenia programów logicznych stanowiących wejście dla systemu ASP. Solver „clasp” z kolei przyjmuje pliki w formacie (języku) aspif, smodels lub podobnym i je przetwarza [24]. Przy zastosowaniu ASP można stworzyć oddzielne pliki, które system ten będzie później analizować. Definiuje się więc problem, rozdzielając konkretny przypadek problemu (**instancję problemu**) od modelu rozwiązania problemu (tzw. **encoding**). Model ten składa się głównie z reguł i korzysta z pliku instancji, w którym zawarte są informacje w postaci faktów przedstawiających obiekty i relacje między obiektami. Na ich podstawie wyszukiwane jest rozwiązanie problemu. Model rozwiązania jest jednolity i uniwersalny w kontekście instancji, czyli stosowalny do wszystkich przypadków problemu (o ile posługują się takim samym, zgodnym sformułowaniem problemu) [5].

**Podstawowymi** elementami języka „gringo” i „clingo” są:

1. Pojęcia – w szczególności stanowiące argumenty innych elementów (liczby całkowite, stałe, łańcuchy znaków symbole zmiennych – np. zmienna  $X$ ) lub funkcje np.  $time(22)$ . [5]
2. reguły – składają się z głowy i/lub ciała, to podstawowe konstrukcje programowania w logice o postaci:

$A0 :- L1, ..., Ln.$

Gdzie  $A0$  to głowa oddzielona symbolem „:-” od ciała reguły zakończona symbolem kropki.

- fakty – reguły składające się tylko z głowy – np.

$ptak(szpak).$

- ograniczenia integralności – reguły składające się tylko z ciała – np.

$:- ptak(P), ma\_skrzydla(P,N), N = 2.$

Charakterystycznym dla ASP konstruktem są tzw. reguły wyboru (choice rules). Reguły te pozwalają sformułować odpowiedź (stabilny model) w wielu kombinacjach. Reguły takie można nazwać regułami wyboru, ponieważ ich działanie polega na wybraniu lub „zgadnięciu” spośród możliwych kombinacji np.: [5]

$\{ p(a) ; q(b) \}.$  - wybierz które spośród podanych uwzględnić w rozwiązaniu

lub przykład bardziej złożony:

$1 \{ cykl(Z,Z1) : zadanie(Z1), Z \neq Z1 \} 1 :- zadanie(Z).$

Charakterystyczną cechą modelu rozwiązania problemu (tzw. *encoding*) jest stosowanie dużej ilości reguł (wyrażeń) oraz podział na sekcje:

- generowanie (*Generate*) – zadaniem reguł w tej sekcji jest wygenerowanie szerokiej gamy możliwych rozwiązań problemu. W związku z tym reguły te w dużej części będą regułami wyboru [5].

- definicja (*Define*) – stanowi przełożenie szczegółowych założeń dotyczących problemu na reguły modelu rozwiązania. Określa pomocnicze zasady i fakty, które korzystają z informacji zawartych w konkretnej instancji problemu umożliwiając. Fakty te są następnie wykorzystane w sekcji test do określenia zgodności z założeniami [5].

- test (*Test*) – stanowi przełożenie wszystkich założonych ograniczeń dotyczących problemu na reguły modelu rozwiązania. W tej sekcji zapisuje się reguły których zadaniem jest odrzucenie tych wygenerowanych rozwiązań, które nie są akceptowalne z perspektywy przestrzeni możliwych, przyjmowalnych rozwiązań problemu. Są to ograniczenia integralności (integrity constraints). Po odrzuceniu nieakceptowalnych rozwiązań, wygenerowane odpowiedzi stanowią tzw. zbiór stabilnych modeli, w którym każde rozwiązanie stanowi możliwe rozwiązanie problemu, opisane w postaci faktów [5].

W przypadku niezgodności z zapisanymi ograniczeniami takie rozwiązanie jest odrzucane (nie będzie obecne w zbiorze stabilnych modeli). Ograniczenia integralności mogą na pierwszy rzut oka być mylące pamiętając, iż dowodzą one, czy dany zapis jest zgodny z podanymi ograniczeniami, czy też nie. Na przykład: w przypadku analizowania energii akumulatora satelity - można założyć, iż zadania mogą być wykonywane przez satelity na orbitach w taki sposób, aby akumulator urządzenia nie uległ wyczerpaniu. W związku z tym ograniczenie integralności może wyglądać następująco:

*:- sat\_energy(Orbit,Action,Value), action(Action,Type), orbit(O), Val < 1.*

gdzie:

*sat\_energy(Orbit, Action, Value)* – relacja określająca ilość *Value* dostępnej energii w momencie rozpoczęcia wykonywania zadania *Action* z orbity *Orbit*

*action(Action, Type)* – relacja między typem zadania (obserwacja, uplink, downlink) a jego numerem

*orbit(O)* – relacja świadcząca o tym, że wartość *O*, to numer orbity

W przypadku zapisu powyższego ograniczenia można odnieść wrażenie, że jest ono błędne, bowiem zgodnie z założeniem, że akumulator nie może się wyczerpać akceptowalna wartość energii w dowolnym momencie w czasie powinna być zawsze większa od 0. Należy więc pamiętać, że reguły te rozumie się na zasadzie „nie może się zdarzyć że”. W tym przypadku – nie może się zdarzyć, że energia akumulatora w momencie wykonywania zadania wyczerpie go (lub wręcz jej ilość nie wystarczy na dokończenie zadania) [5].

- prezentowanie (*Display*) – wpisane w tej sekcji dyrektywy w formie np.:



*#show sat\_energy/3.*

*#show orbit/1.*

Mając za zadanie zawężanie ilości informacji związanymi z wyszukanymi rozwiązaniami do potrzebnych atomów, czyli odrębnych faktów (w powyższym przypadku do wszystkich relacji *sat\_energy* o trzech argumentach oraz relacji *orbit* o jednym argumentem)

Tego typu podejście, które polega na podzieleniu modelu na odrębne sekcje wzorcem „Generate-Define-Test-Display” [5].

System „clingo” posiada również szereg dodatkowych opcji uruchamiania np.

- uruchamianie wielowątkowe
- zatrzymywanie w momencie odnalezienia *n* modeli
- podawanie statystyk i szczegółowych danych dotyczących czasu pracy i przebiegu programu po zakończeniu pracy

Szczegółowe informacje w zakresie wszystkich dostępnych funkcji, zasad działania wraz z przykładami implementacji opisane są w dokumentacji wydanej przez autorów systemu [24].

Technologię ASP obecnie wykorzystano także w biznesie. Powstały rozmaite implementacje na bazie tej technologii i powiązanych narzędzi dodające i łączące w sobie dodatkowe możliwości, takie jak możliwość komunikacji z zewnętrzną bazą danych, integracja z językiem Java (JASP), albo zintegrowane z ASP środowisko deweloperskie (DLV), które to ułatwiają zastosowanie ASP w nieprzystosowanych systemach (np. zaprojektowanych i napisanych obiektowo). Umożliwiło to włączenie ASP do oprogramowania przemysłowego, zajmującego się zarządzaniem zasobami ludzkimi w przedsiębiorstwach lub w systemach e-commerce podmiotów świadczących usługi turystyczne. Stwierdzono, iż zastosowanie ASP pozwala na obsługiwanie zwykle skomplikowanej logiki biznesowej przedsiębiorstwa stosunkowo niewielkim nakładem środków przeznaczonych na implementację (zmniejszenie ilości oprogramowania w porównaniu do implementacji w paradygmacie imperatywnym). Dodatkowymi zaletami związanymi z utrzymywaniem oprogramowania w przypadku tej technologii jest czytelność, łatwa skalowalność i elastyczność kodu (łatwe modyfikowanie, z którym nie wiąże się awaryjność oprogramowania) [24].

Świadczy to o możliwości wykorzystania tego narzędzia do rozwiązywania problemów związanych z zarządzaniem, optymalizowaniem i planowaniem wewnątrz przedsiębiorstw (procesami produkcyjnymi, zasobami ludzkimi), są to typy problemów charakteryzujące się zwykle dużą ilością wykonywanych przy ich rozwiązywaniu obliczeń np. przeanalizowania w pewnym momencie wszystkich możliwych kombinacji. Systemy ASP w rozwiązywaniu problemów o złożoności wykładniczej.

Systemy wnioskowania ASP ze swymi możliwościami do efektywnego generowania dużej ilości wiedzy umożliwiają rozwiązywanie problemów kombinatorycznych polegających w swej istocie na

przeszukiwaniu przestrzeni rozwiązań, są więc w szczególności przystosowane do rozwiązywania problemów NP-trudnych, przy czym wygenerowane rozwiązanie jest (charakterystycznie dla ASP) w postaci zbioru stabilnych modeli [15].

#### **<#>złożoność wykładnicza, problem kombinatoryczny (duża złożoność obliczeniowa)</#>**

Stosując programowanie Answer Set Programming wygenerowany wynik (rozwiązanie problemu) zwykle przedstawia się w formie dużej ilości stabilnych modeli (poprawnych rozwiązań). Często istnieje potrzeba uzyskania wyniku najlepszego pod względem jakiegoś przyjętego kryterium, np. zmaksymalizowanie zysku - można stosować wtedy optymalizację wygenerowanych wyników. Wygenerowany wynik może nie mieć również żadnych odpowiedzi, co również jest bardzo istotną informacją. Przy badaniu niektórych problemów formułuje się je w taki sposób, aby otrzymać odpowiedź na przykład na pytanie: czy istnieje w ogóle jakiekolwiek rozwiązanie dla postawionego problemu. Są to zwykle problemy trudno-rozwiązywalne. W tym przypadku system ASP dysponuje pewnego rodzaju wewnętrznymi mechanizmami, które mają na celu zmniejszanie ilości obliczeń (a więc i czasu potrzebnego na ich przeprowadzenie) do uzyskania jakiegokolwiek poprawnej odpowiedzi tzw. „time to solution”. Na przykład w przypadku problemów NP-trudnych opartych na kombinatorycznym przeszukiwaniu w istocie jedynym wyjściem jest znalezienie rozwiązania w przestrzeni skończonej ilości przypadków. W związku z tym, system ASP jest dobrze przystosowanym narzędziem podczas prób rozwiązywania problemów kombinatorycznych. W przypadku tego typu problemów model rozwiązania problemu zapisany w formie programu logicznego (encoding) będzie kładł nacisk głównie na dwa typy reguł:

- reguły wyboru – ich zadaniem będzie kombinatoryczne generowanie przestrzeni rozwiązań
- ograniczeń integralności – mających za zadanie filtrowanie poprawnych rozwiązań z przestrzeni wszystkich wygenerowanych [5].

## 2. Charakterystyka problemu

Pomimo dużej ilości satelitów obserwacyjnych, które współcześnie orbitują Ziemię, ilość zadań i wymagań stawianych w zakresie obserwacji zjawisk pogodowych, wód i lądów stwarza konieczność zaplanowania obserwacji w formie szeregu zadań obserwacyjnych do wykonania przez satelitę, który w efekcie ma skutecznie wypełniać swoje praktyczne przeznaczenie dostarczając dane w dobrej jakości, w sposób efektywny, jednocześnie operując pod presją czasu, prawdopodobieństwa wystąpienia nagłych lub nieprzewidywalnych zdarzeń i ograniczeń różnego pochodzenia (wynikających z warunków naturalnych, czasu, będącej efektem zastosowanej technologii itd.).

### **Problem harmonogramowania**

Problem harmonogramowania w sensie ogólnym polega na odnalezieniu takiego, optymalnego układu zadań do wykonania w czasie, który zapewnia spełnienie konkretnych, ściśle zdefiniowanych założeń lub ograniczeń.

Potrzeba stworzenia harmonogramu, który pozwala na wykonanie pewnej pracy w minimalnym czasie przy optymalnym wykorzystaniu zasobów pojawia się wszędzie tam gdzie jest potrzeba wykonania zestawu zadań i dochodzi do planowania przebiegu ich wykonywania przy limitowanych zasobach różnego rodzaju (pieniędzy, pracowników, urządzeń itd.) i istnieje potrzeba optymalizacji np. minimalizacja czasu wykonania lub zużycia zasobów. Ponadto zwykle dodatkową trudnością są pewne odgórne ograniczenia, których nie można wyeliminować. W związku z tym, przeprowadzenie harmonogramowania wymaga kombinatorycznego przeanalizowania wszystkich możliwości (dokonanie przeszukiwania, podobnie do problemu wędrownego sprzedawcy), co powoduje, że problem charakteryzuje się złożonością  $n!$  (w najprostszej wersji). Algorytm o czasie wykładniczym trudno stosować w praktyce, ze względu na długi czas wykonywania już przy niewielkim rozmiarze danych – zwykle jest to rozwiązanie nieprzydatne w użyciu, ponieważ zdadne do użycia rozwiązanie można uzyskać przy bardzo niewielkim rozmiarze danych, warunek taki zwykle nie jest spełniony, a w przypadku przeciętnym rozwiązanie nie jest efektywne, czyli czas potrzebny na uzyskanie rozwiązania wyklucza możliwość praktycznego zastosowania. Harmonogramowanie to problem NP-trudny. „Gdy problem optymalizacyjny jest NP-trudny (...), nie istnieje algorytm o wielomianowym czasie działania znajdujący najlepsze rozwiązanie, chyba że  $P = NP$  [6] <#>CYTAT</#>. Jest to jeden z problemów milenijnych. Natomiast nie tylko nie rozstrzygnięto, że  $P \neq NP$ , ale istnieją również dowody na to, że nie można dowieść  $P \neq NP$ ”. Ponadto, harmonogramowanie jest czynnością polegającą na uporządkowaniu obiektów w pewnej kolejności, co powoduje, że jest to problem redukowalny do cyklu Hamiltona (problemu TSP) lub problemu CSP. Z uwagi na to, że są to zagadnienia NP-trudne, analizowany problem

harmonogramowania również musi być problemem NP-trudnym. Sedno problemu w różnych przypadkach harmonogramowania jest takie samo, inne są natomiast ograniczenia, które należy uwzględnić, sposób optymalizacji lub dostępne zasoby posiadają inny charakter. Przykłady konkretnych przypadków, przy których podejmowano już szczegółowe analizy to harmonogramowanie warsztatu (tzw. „job shop scheduling”), czy tzw. problem pielęgniarek (nurse scheduling problem). Badania w zakresie tych przypadków podejmowano we wczesnych latach 50-tych XX w. [16]. W związku z tym, że problem należy do grupy problemów NP-trudnych i pojawia się w codzienności bardzo często (także w przedsiębiorstwach i zarządzaniu dużymi projektami) stanowi bardzo dobrą podstawę do testowania i badania algorytmów w różnych wersjach, podejmujących próbę rozwiązywania harmonogramowania.

<#>Można też opisać problem jako Constraint Satisfaction Problem</#>

## 2.1 Przegląd dotychczasowych badań

### Harmonogramowanie w podstawowej wersji

Problem harmonogramowania w klasycznych odsłonach (harmonogramowanie pracy warsztatu lub problem pielęgniarek) był wielokrotnie analizowany. Samo harmonogramowanie jest dosyć powszechnym problemem stąd zaproponowane zostały różne metody - algorytmy zachłanne oraz heurystyki umożliwiające uzyskanie praktycznego rozwiązania. W dużej części są to heurystyki i strategie ewolucyjne z których interesujące są między innymi: [18]:

- zoptymalizowane heurystyczne przeszukiwanie w połączeniu ze strategią *branch and bound*

Stworzono zoptymalizowaną heurystykę przeszukującą wewnętrznie łączącą GRASP (*Greedy Randomised Adaptive Search Procedure*) ze strategią *branch & bound* [20] dla harmonogramowania pracy maszyn. Jest to hybryda heurystyki działającej na bazie lokalnego przeszukiwania z algorytmem dokładnym.

Harmonogramowanie obserwacji satelitarnych jest przypadkiem harmonogramowania, gdzie dodatkowe ograniczenia wynikają między innymi z orbitowania satelity nad powierzchnią Ziemi. Naziemny tor ruchu satelity w połączeniu z dokładnym położeniem (współrzędnymi) obiektu obserwowanego na powierzchni planety oraz dodatkowymi możliwościami satelity na przykład związanymi z dostępnym kątem widzenia kamery pokładowej powodują ograniczenie w postaci okna czasowego widoczności. Rozpatrując harmonogramowania obserwacji satelitarnych podjęto próby rozwiązywania tego problemu stosując między innymi [21]:

- algorytmy zachłanne,
- algorytmy genetyczne,
- ewolucja różnicowa,
- optymalizacja mrówkowa,

- optymalizacja stadna cząstek,
- symulowane wyżarzanie,
- programowanie ograniczeniami,
- programowanie całkowitoliczbowe mieszane,
- przeszukiwanie np. „tabu search”,
- strategia „branch and bound”

Są wśród nich algorytmy dokładne i metody przybliżone, przy czym te pierwsze są efektywne dla problemów mniejszych rozmiarów [21]. Wybrane metody opisano poniżej.

### **Programowanie ograniczeniami**

W tej metodzie zaproponowano podejście GREAS (Generic Resource, Event, and Activity Scheduler) łączącą elastyczne przypisywanie zasobów do zadań w oknach czasowych widoczności oraz zastosowanie solvera „IBM ILOG scheduler” do analizy spełnialności wszystkich założonych ograniczeń. Metoda ta zmniejsza przestrzeń poszukiwań w trakcie przypisywania kolejnych zmiennych przez zastosowanie techniki propagacji ograniczeń (każde ograniczenie ma konkretne konsekwencje przy uporządkowywaniu zadań i przypisywaniu zasobów). W pierwszej kolejności zamodelowano problem przekształcając charakterystyczne wielkości (zadania, okna czasowe widoczności, start zadania i czas jego wykonywania itd.) do struktur zmiennych o pewnych akceptowalnych wartościach. Następnie dochodzi do szczegółowej analizy zadań, zasobów i ograniczeń w poszukiwaniu efektywnego harmonogramu [19]. Ostatecznie przebieg harmonogramowania dla danego przyjętego modelu wygląda w następujący sposób [19]:

1. Wybór zmiennej i jej akceptowalnej wartości. Zwykle wykonywanie jest heurystyczne przeszukiwanie z możliwych dostępnych wartości.
2. Propagacja wpływu wyboru na pozostałe zmienne w modelu - analiza wpływu przypisania tej wartości na całość harmonogramu i korekty innych wielkości
3. Powtórzenie kroków 1 i 2, aż do momentu, aż wszystkie zmienne modelu będą miały przypisane wartości zgodne z ograniczeniami. Jeśli podczas propagacji okaże się, że nie ma możliwości przypisania poprawnej wartości do zmiennej algorytm cofa się i powtarza czynności wybierając inną wartość z dostępnego zakresu.

Czas startu i zakończenie wykonywania zadania wybierany jest w sposób heurystyczny. Oznacza to, że z całego dostępnego okna czasowego dla danego zadania wybierany jest ten okres jego wykonywania, który jest najbardziej optymalny z perspektywy dalszej analizy (przypisywania kolejnych zadań i zasobów). W przypadku zadań przy których rolę odgrywa kilka zmiennych i ograniczeń, na przykład odsyłanie danych do stacji naziemnej (możliwe okna czasowe widoczności stacji i satelitów, które w danym momencie oczekują na komunikację z nią) metoda ta uwzględnia poszukiwanie optymalnej kolejności łączenia się z danymi satelitami w czasie. Ponadto istnieje

także możliwość harmonogramowania z uwzględnieniem dzielenia czasu wykonywania zadań na kilka części. Przerywanie wykonywania zadań i wykonywanie ich przez satelitę w kilku krokach zwiększa do ilość możliwych kombinacji, ale pozwala również na uzyskanie bardziej optymalnego podziału zasobów. W przypadku zadań polegających na odnawianiu zasobów, czyli odsyłaniu danych (odzyskiwanie pamięci) stosowane są heurystyki polegające na minimalizacji tych czynności przy jednoczesnym zachowaniu pełnej możliwości wykonywania zadań obserwacyjnych konsumujących pamięć. Jest to bezpośrednio związane z ostatecznym harmonogramem, jednak metoda pozwala też na odgórne zdefiniowanie ilości zadań odnawiających zasoby i wygenerowanie niekoniecznie optymalnego, ale spójnego harmonogramu przy uwzględnieniu tych odgórnych ustaleń [19].

Adaptacyjne symulowane wyżarzanie z dynamicznym grupowaniem zadań

ASA-DTC (*Adaptive Simulated Annealing with Dynamic Task Clustering strategy*) jest to kombinacja różnych heurystyk i metod (w tym np. symulowane wyżarzanie, „tabu search”), w której główną rolę odgrywają heurystyczne harmonogramowanie i dodatkowe zwiększanie efektywności poprzez zgrupowanie zadań. Zauważono, że konwencjonalne symulowane wyżarzanie w rozwiązywaniu harmonogramowania może mimo wszystko zwracać rozwiązanie mało optymalne, a uzyskanie wyników dokładniejszych wiąże się z dużym wzrostem ilości czasu potrzebnego na zakończenie procesu harmonogramowania. Grupowanie bazuje na tym, że niektóre zadania obserwacyjne docelowo wykonywane są na niewielkiej powierzchni, zatem okna czasowe widoczności i położenie sensorów dla tych zadań są takie same. W związku z tym bardziej efektywnym staje się wykonanie kilku zadań traktując je jako pojedyncze. Grupowanie zadań znacząco zwiększa zatem efektywność generowania harmonogramu, jednak statyczne grupowanie bezpośrednio przed przeprowadzeniem harmonogramowania uniemożliwia określanie maksymalnie optymalnego porządku wykonywania zadań (podczas harmonogramowania może okazać się, że inne zgrupowanie niektórych zadań jest efektywniejsze). Grupowanie w sposób dynamiczny wbudowane jest w proces przeszukiwania sąsiednich punktów (symulowane wyżarzanie) i znacząco poprawia niedogodność występującą w przypadku grupowania statycznego w efekcie uzyskując lepszą jakość harmonogramowania [21].

W metodzie tej ważną rolę odgrywają:

- adaptacyjne sterowanie wartością temperatury i zoptymalizowany dobór sąsiedztwa w procedurze symulowanego wyżarzania
- przeszukiwanie „tabu”, którego zadaniem jest spowodowanie pomijania wielokrotnej analizy tych samych kombinacji przy rozwiązywaniu

Docelowo, metoda dąży do maksymalizacji efektu (jakości) harmonogramowania, co przedstawia się formalnie jako:

$$\max \sum_{i=1}^N \sum_{j=1}^M x_{ij} \cdot p_i$$

$$x_{ij} = \begin{cases} 0 & \text{– wykonanie zadania i niezaplanowane na orbicie j} \\ 1 & \text{– wykonanie zadania i zaplanowane na orbicie j} \end{cases}$$

**gdzie:**

$j = 1, 2 \dots M$  – zbiór analizowanych orbit

$i = 1, 2 \dots N$  – zbiór zadań obserwacyjnych

$x_{ij}$  – zmienna decyzyjna określająca zaplanowanie wykonania zadania obserwacyjnego  $i$  z orbity  $j$

$p_i$  – priorytet zadania obserwacyjnego

Metoda ASA-DTC określa następujące ograniczenia dla harmonogramowania

1. Każde zadanie może być wykonane maksymalnie raz.
2. Pomiędzy wykonywanymi zadaniami obserwacyjnymi wymagany jest pewien przedział czasu wykorzystany na kalibrację sensora i ładowanie energii, która zużyta zostanie przez sensor w trakcie kolejnego zadania – dochodzi do przerwy w wykonywaniu obserwacji.
3. Ładowanie i kalibracja sensora wykonywana jest między każdymi kolejnymi zadaniami.
4. Energia urządzenia satelitarnego nie może się wyczerpać (zużycie nie przekracza akumulowanej pojemności).
5. Dostępna pamięć komputera satelity nie może się zapełnić (zapełnienie nie przekracza dostępnej pojemności) [21].

### 3. Model DLP w rozwiązywaniu

## harmonogramowania obserwacji satelitarnych

W tym rozdziale dokonano charakterystyki modelu DLP stworzonego na potrzeby badań związanych z harmonogramowaniem obserwacji satelitarnych. W szczególności opisano strukturę stworzonego modelu DLP, sposobu jego działania oraz niezbędnych wyjaśnień.

Problem harmonogramowania obserwacji satelitarnych polega ostatecznie na przypisaniu zadań obserwacyjnych w odpowiedniej kolejności do dostępnych zasobów, czyli satelitów znajdujących się na orbitach biorąc pod uwagę wymagane ograniczenia bezpośrednio wynikające z możliwości tych urządzeń i oczekiwanych rezultatów. Harmonogram musi uwzględniać zatem przyjęte z góry założenia, które wynikają z praw fizyki, konstrukcji i sposobu działania urządzenia, organizacji pracy i innych, często trudnych do przewidzenia wyzwań. Wygenerowany harmonogram nadawał się będzie wtedy do wyciągnięcia praktycznych wniosków dotyczących zastosowania w rzeczywistych warunkach.

Podczas badań wykorzystano zintegrowany system ASP umożliwiający badanie harmonogramowania na różne sposoby, w zależności od sposobu sformułowania problemu (modelu). Problem w podstawowej wersji dotyczy określenia dokładnego, optymalnego przypisania dostępnych zasobów do z góry ustalonych zadań oczekujących na wykonanie. Dostępnymi zasobami są w tym przypadku orbitujące satelity, które różnią się między sobą parametrami technicznymi satelitów na nich osadzonych i możliwością obserwacji poszczególnych zadań (widocznością zadania z pozycji danej satelity). W przypadku ograniczeń związanych z zasobami satelity mowa jest o:

- ilości dostępnej pamięci komputera satelity,
- rozmiaru zajmowanej pamięci przez dane z obserwacji, czyli kosztu pamięciowego wykonania zadania obserwacyjnego,
- pojemności akumulatora satelity,
- kosztu energetycznego wykonywania zadań (poboru energii podczas wykonywania zadań),
- szybkości ładowania akumulatora za pomocą ogniw słonecznych,
- czasu potrzebnego na wykonywanie wszystkich rodzajów zadań,

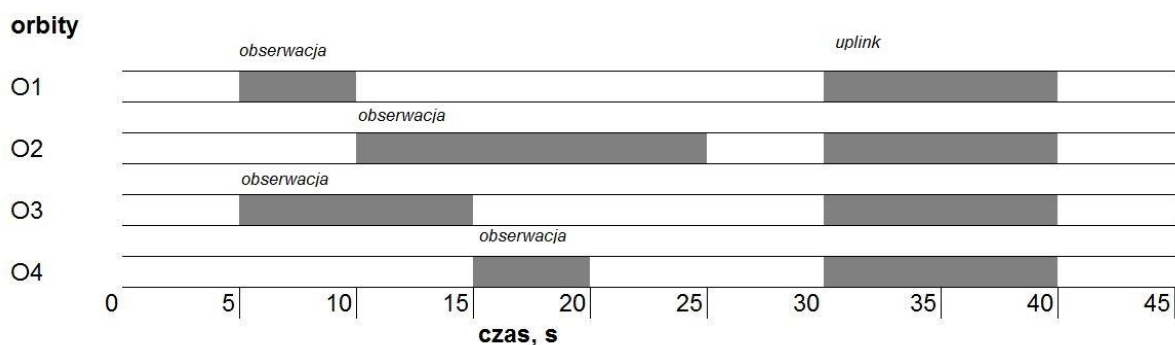
Zadania oczekujące na wykonanie są rozłożone w czasie - znany jest czas, w którym każde zadanie może być wykonane na podstawie ilości sekund potrzebnych na przemieszczenie się satelitów po orbicie na pozycję nad obiektem obserwowanym (lub stacjami komunikacyjnymi: kontrolnymi, naziemnymi). Biorąc pod uwagę charakter problemu obserwacji i tor ruchu satelitów po orbitach konieczne należy uwzględnić przedział czasowy, w którym dany obszar jest widzialny przez



urządzenie z pozycji na orbicie, czyli okna czasowe widoczności zadań. W rzeczywistości rozkład zadań oczekujących na wykonanie nie jest równomiernie rozłożony w czasie. Innymi słowy, różne obszary na powierzchni planety charakteryzują się różną gęstością zadań. Powoduje to, że zadania w pewnym sensie konkurują między sobą o satelity, które nie koniecznie są w stanie wykonać wszystkie zadania oczekujące na wykonanie. Dodatkowo warunkiem akceptowalności harmonogramu jest przypisanie wszystkich zadań awaryjnych (to jedno z ograniczeń). W związku z tym dobór przypisywanych zadań jest jeszcze bardziej restrykcyjny.

### Poglądowy przykład przypisania zadań:

- harmonogramowanie odbywa się dla 4 orbit
- zadania 1-4 są zadaniami obserwacyjnymi
- zadanie 5 do zadanie typu *uplink*
- czas obserwacji wynosi 5 s dla satelitów na orbitach O1 i O4, 10 s dla satelity na orbicie O2, i 15 s dla satelity na orbicie O3
- czas zadania *uplink* na wszystkich orbitach trwa 10 s
- widoczność zadań to: obserwacja 1 – widoczne na O1, obserwacja 2 – widoczne na O2, obserwacja 3 – widoczne na O3, obserwacja 4 – widoczne na O4, *uplink* (5) – widoczne na orbitach O1 do O4.
- okna czasowe widoczności zadań to: obserwacja 1 – [5;13], obserwacja 2 – [10;30], obserwacja 3 – [5;18], obserwacja 4, *uplink* 5 – [30;45]



**Rysunek 3.** Poglądowy schemat przypisania zadań do orbit

W opisywanym przykładzie poprawnym harmonogramem może być przedstawiony powyżej diagram przebiegu wykonywania zadań.

- obserwacja 1 – przypisano do wykonania na jedynej orbicie, z której jest ono widoczne O1, wykonanie (okno czasowe wykonywania zadania) zaplanowano na czas [5;9]

- obserwacja 2 – przypisano do wykonania na jedynej orbicie, z której jest ono widoczne O2, wykonanie zaplanowano na czas [10;24]
- obserwacja 3 – przypisano do wykonania na jedynej orbicie, z której jest ono widoczne O3, wykonanie zaplanowano na czas [5;14]
- obserwacja 4 – przypisano do wykonania na jedynej orbicie, z której jest ono widoczne O4, wykonanie zaplanowano na czas [15;19]
- uplink 5 – przypisano do wykonania na wszystkich orbitach O1 – O4, wykonanie zaplanowano na każdej z nich w czasie [30;39]

### 3.1 Definicja modelu

Problem harmonogramowania obserwacji satelitarnych można przedstawić na wiele różnych sposobów. Na podstawie przeglądu dotychczasowych badań i prób rozwiązania problemu można stwierdzić, że dotychczasowe zetknięcia się z tym problemem charakteryzowały się mniej lub bardziej szczegółowym sformułowaniem problemu, w związku z czym stosowano mniej lub więcej parametrów pracy satelity takich jak pamięć, energia, sposób i częstotliwość komunikacji ze stacjami, możliwość zmiany pozycji satelity w stosunku do trajektorii orbity za pomocą silników pokładowych lub zmiana kąta nachylenia urządzenia obserwacyjnego lub anten komunikacyjnych. Ponadto w różny sposób rozważano ograniczenia czasowe w wykonywaniu zadań. Okna czasowe określano z góry lub umożliwiano wyznaczanie rozkładu (kolejności) i możliwości wykonywania zadań na podstawie prędkości ruchu urządzenia satelitarnego i rodzaju orbity na której zostało umieszczone. Stosowano tutaj różnego rodzaju mapy, współrzędne zaplanowanych zadań (punktów na powierzchni planety) w celu przeprowadzenia pewnego rodzaju symulacji ruchu satelity i pozyskania potrzebnych danych na tej podstawie. Koniecznością związaną z klasą złożoności problemu stało się też szukanie sposobu na optymalizację samych obliczeń podczas harmonogramowania. W tym przypadku badacze stosowali heurystyki, różnego rodzaju techniki ewolucyjne lub metody preprocesowania rzeczywistego problemu umożliwiającego pewną optymalizację obliczeń.

Na potrzeby badań zastosowano indywidualne podejście do problemu. Przede wszystkim charakterystykę urządzeń satelitarnych, orbit i układ zadań sprowadzono do liczb całkowitych (można to nazwać programowaniem całkowitoliczbowym, albo mieszano-całkowitoliczbowym). W ten sposób można zastosować system ASP i zaprojektować model rozwiązania. W związku z tym większość założeń po przełożeniu na język programu logicznego ma postać liczb i związków między tymi liczbami, pod którym kryje się faktyczny opis problemu. Staje się on logiczną całością w momencie zdefiniowania wewnątrz niego obiektów i relacji między tymi obiektami, np.:

*orbit(1).*

(a)

*satellite(orbit(1), aqua).*

(b)

lub

*orbit(1).*

(a)

*satellite(1, aqua).*

(b)

Należy rozumieć jako:

(a) - istnieje orbita 1

(b) - na orbicie 1 operuje satelita „aqua”.

Na początku prac zarysowano ramy definicji problemu jednoznacznie określając wszystkie jego założenia modelując tym samym rzeczywisty charakter pracy urządzeń satelitarnych na orbitach, typy istniejących zadań planowanych do wykonania, sposób harmonogramowania. Założenia określają również szczegółowe ograniczenia wynikające z procesu harmonogramowania, w tym:

- ograniczenia techniczne dotyczące satelitów operujących na orbitach – pamięć komputera satelity, pobór energii podczas prac, czas potrzebny na wykonanie zadania
- ograniczenia w harmonogramowaniu poszczególnych typów zadań (mogą lub muszą być wykonane)

Wszystkie założenia przedstawiono poniżej:

1. Istnieje zdefiniowana ilość orbit.
2. Na każdej orbicie znajduje się jeden satelita.
3. Zadania wykonywane są przez satelity znajdujące się na orbitach.
4. Wykonywanie zadania musi odbyć się w dostępnym oknie czasowym jego widoczności.
5. Istnieją 3 typy zadań:
  - obserwacyjne,
  - downlink (dostarczanie danych na powierzchnię Ziemi),
  - uplink (odbieranie danych z powierzchni Ziemi).
6. Zadanie obserwacyjne może być wykonane raz, z dowolnej orbity, z której wykonanie go jest możliwe.
7. Downlink **może** być wykonany przez dowolną ilość satelitów z orbit z których wykonanie go jest możliwe.
8. Uplink **musi** być wykonany przez wszystkie satelity z orbit z których wykonanie go jest możliwe.
9. Zadanie obserwacyjne charakteryzuje się priorytetem. Im wyższy priorytet, tym **ważniejsze** jest wykonanie zadania.
10. Awaryjne zadanie obserwacyjne **musi** być obligatoryjnie wykonane.

11. Satelita charakteryzuje się określoną ilością dostępnej **pamięci**.
12. Pamięć komputera satelity zajmowana jest przez dane z wykonanych zadań **obserwacyjnych**.
13. Koszt pamięciowy wykonania każdego zadania jest **jednakowy** dla tego samego satelity, lecz poszczególne satelity mogą charakteryzować się **różnym** kosztem **pamięciowym** wykonywania zadania.
14. Satelita o ilości wolnej pamięci mniejszej niż koszt pamięciowy wykonania zadania w danym momencie **nie może** wykonywać zadań obserwacyjnych, aż do momentu **zwolnienia** pamięci.
15. Pamięć komputera satelity jest całkowicie **zwalniana** po wykonaniu odsyłania danych do stacji naziemnej (**downlink**).
16. Czas odsyłania danych (downlink) jest **jednakowy** dla tego samego satelity, lecz poszczególne satelity mogą charakteryzować się **różnym** czasem wysyłania danych.
17. Satelita charakteryzuje się jednoznacznie określoną ilością dostępnej **energii** akumulatora.
18. Dostępna energia satelity zmniejsza się przy wykonywaniu zadań obserwacyjnych o wartość kosztu energetycznego wykonania zadania.
19. Koszt energetyczny wykonania każdego zadania jest **jednakowy** dla tego samego satelity, lecz poszczególne satelity mogą charakteryzować się **różnym** zużyciem energii przy wykonywaniu zadania.
20. Satelita o ilości dostępnej energii mniejszej niż koszt energetyczny wykonania zadania w danym momencie **nie może** wykonywać zadań aż do momentu **odnowienia** energii akumulatora o wartość równą kosztowi energetycznemu wykonania zadania dla satelity.
21. Energia satelity jest **odnawiana** proporcjonalnie do upływu czasu. Wartość generowanej energii jest **stała** lecz poszczególne satelity mogą charakteryzować się **różną** wartością generowaną w jednostce czasu. Generowanie odbywa się automatycznie podczas operowania satelity (przy pomocy ogniw słonecznych).
22. Czas komunikacji kontrolnej (uplink) jest **jednakowy** dla tego samego satelity, lecz poszczególne satelity mogą charakteryzować się **różnym** czasem komunikacji.
23. Żadne okno czasowe wykonywania zadania przed tego samego satelitą **nie może** nakładać się na drugie (tylko jedna czynność wykonywana jest w danym czasie).
24. Harmonogram powinien obejmować stały przedział czasu operowania satelitów (planowanie harmonogramu np. na 50 lub 100 minut wprzód).
25. Harmonogram obserwacji satelitarnych jest tym **lepszy**, im wyższa jest wartość **jakości** harmonogramu.

26. Jakość harmonogramu określona jest sumą priorytetów z **przypisanych** do wykonania zadań wg wzoru:

$$q = \sum_{i=1}^n p_i \cdot x_i$$

**gdzie:**

q – jakość harmonogramu

n – całkowita ilość zadań obserwacyjnych

p<sub>i</sub> – priorytet zadania

x<sub>i</sub> – wartość binarna {0,1} opisująca stan przypisania zadania (1 jeśli zadanie przypisano do którejś z orbit, 0 jeśli zadanie pominięto na rzecz przypisania innych)

### 3.2 Instancja problemu

Instancja problemu musi przede wszystkim uwzględniać przedstawione wcześniej założenia dotyczące kryteriów i sposobu określania harmonogramu prac obserwacyjnych satelitów. Instancja problemu zawiera reguły w postaci faktów (wiedzę), które oznaczają następująco:

*orbit(1). orbit(2). orbit(3).*

- określa rozpatrywane orbity (czyt. 1 jest orbitą)

*satellite(1,calipso). satellite(2,aqua). satellite(3,cloudsat).*

- określa nazwy satelitów zajmujących orbity (czyt. orbita 1 zajmowana jest przez satelitę landsat)

*action\_type(observe).*

- określa, rozpatrywane typy zadania (np. obserwacja - czyt. obserwacja to typ zadania)

*energy\_gen(1,8).*

- określa ilość jednostek energii generowanych przez satelity na kolejnych orbitach (czyt. satelita na orbicie 1 generuje 8 jednostek energii w jednostce czasu)

*energy\_storage(1,500).*

- określa pojemność akumulatora satelity na kolejnych orbitach (czyt. satelita na orbicie 1 dysponuje akumulatorem o pojemności 500)

*energy\_use(1,40).*

- określa koszt (ilość zużytych jednostek energii) wykonania zadania przez satelity na kolejnych orbitach (czyt. satelita na orbicie 1 zużywa 40 jednostek energii przy wykonywaniu zadania)

*memory\_storage(1,800).*

- określa pojemność pamięciową komputera satelity na kolejnych orbitach (czyt. komputer satelity na orbicie 1 dysponuje pojemnością 800 jednostek pamięci)

*memory\_use(1,50).*

- określa koszt (ilość jednostek pamięci) wykonania zadania przez satelity na kolejnych orbitach (czyt. wykonanie zadania obserwacyjnego kosztuje 50 jednostek pamięci komputera satelity)

*sat\_action\_time(1,downlink,12).*

- określa czas (ilość jednostek czasu) potrzebny na wykonania zadania danego typu (obserwacja, downlink i uplink) przez satelity na kolejnych orbitach (czyt. satelita z orbity 1 na wykonanie zadania downlink potrzebuje 12 jednostek czasu)

*sat\_energy(1,0,500).*

- określa stan akumulatora (ilość jednostek energii) na początku harmonogramowania przez kolejne satelity (czyt. akumulator satelity z orbity 1 w momencie startu – zadanie 0, ma dostępne 500 jednostek energii)

*sat\_memory(1,0,0).*

- określa stan pamięci (ilość zajętych jednostek pamięci komputera kolejnych satelitów) na początku harmonogramowania czyt. komputer satelity z orbity 1 w momencie startu – zadanie 0, ma zajęte 0 jednostek energii)

*action(1,observe).*

- wiąże kolejne zadania z 3 dostępnymi typami (czyt. zadanie 1 to obserwacja)

*action\_window(1,6,24).*

- określa rzeczywiste okna czasowe w których możliwe jest wykonywanie kolejnych zadań przez satelity na orbitach (czyt. zadanie 1 ma okno czasowe widoczności w przedziale czasowym [6;24])

*visible(1,1,1).*

*visible(1,2,0).*

- określa możliwość wykonania (widoczność) kolejnych zadań z pozycji w przestrzeni na kolejnych orbitach (np. zadanie 1 jest widoczne z orbity 1, ale nie jest widoczne z orbity 2)

*emergency\_task(2).*

- definiuje obserwacyjne zadania awaryjne (czyt. 2 jest zadaniem awaryjnym i musi zostać wykonane, jeśli harmonogram ma być akceptowalny)

*priority(2,50).*

- definiuje priorytety poszczególnych zadań obserwacyjnych (czyt. zadanie obserwacyjne 2 ma priorytet o wartości 50)

Przykładowa instancja problemu ukazana jest w załączniku B.

### 3.3 Opis modelu DLP

Model rozwiązania problemu (*encoding*) stworzono według podziału generowanie, definicja, test, prezentowanie. Rozszerzono natomiast to podejście, ponieważ można zauważyć, że dodatkowo należy przekazać systemowi ASP jaka wartość powinna podlegać ocenie i zostać

zmaksymalizowana (lub zminimalizowana) podczas wykonywania obliczeń. Dotyczy to optymalizacji rozwiązania, czyli poszukiwania najlepszego harmonogramu. Dodano więc krótką sekcję optymalizacji oraz wyrażenia (instrukcje) dotyczące wyświetlenia wyników. Struktura modelu jest więc podzielona na 4 sekcje.

Dla poszczególnych typów zadań zostały utworzone osobne reguły, mające na celu generowanie możliwych rozwiązań problemu. Zastosowano reguły wyboru, których zadaniem jest wygenerowanie możliwego przypisania zadań określonych w instancji problemu do wykonania na orbicie o danym numerze oraz wybraniu przedziału czasu w obrębie dostępnego okna czasowego, w którym dane zadanie ma być wykonane. Ostatecznie przełożono tutaj na model DLP założenia dotyczące wykonywania zadań a dokładniej:

- zadanie obserwacyjne może być wykonane z jednej z orbit, z której jest to możliwe,
- *downlink* do stacji naziemnej może być wykonywany przez dowolną ilość satelitów z orbit, z których jest to możliwe,
- *uplink* ze stacją kontrolną musi być wykonany przez wszystkie satelity znajdujące się na orbitach, z których jest to możliwe,
- wykonywanie zadania musi odbyć się w obrębie okna czasowego jego widoczności, czyli w czasie, kiedy satelita znajduje się w przestrzeni nad obszarem na powierzchni planety związanym z wykonywaniem zadania.

W bloku definicji utworzono część modelu reprezentującą wymagania związane z parametrami technicznymi satelitów znajdujących się na kolejnych orbitach. Pojawiają się tutaj reguły dotyczące pamięci i energii, którymi dysponują urządzenia. Dodatkowo umieszczono także reguły dotyczące ilości wykonanych zadań i typów zadań z kolejnych orbit (przez operujące urządzenia).

W części zawierającej ograniczenia integralności umieszczono reguły mające na celu odrzucenie nie akceptowalnych wyników. Znajdują się tutaj wszystkie ograniczenia zawarte w podstawowych założeniach dotyczących problemu, głównie dotyczy to zadań awaryjnych lub przekroczenia ilości pamięci lub wyczerpania akumulatora któregoś satelity do którego doszłoby w przypadku zastosowania danego harmonogramu w rzeczywistości. Takie wyniki muszą być odrzucone.

Optymalizacja w przypadku problemu harmonogramowania obserwacji satelitarnych, gdzie kolejne zadania obserwacyjne mają przypisany priorytet w postaci liczby całkowitej polega na maksymalizacji wartości jakości przypisania zadań do orbit (wzór do oceniania jakości harmonogramu podano w rozdziale 3.1).

**<#>odnosnik</#>**

Każde z możliwych rozwiązań (stabilnych modeli) niesie ze sobą informacje, z których wynika min. przypisanie zadań do ponumerowanych orbit oraz rozkład wykonywania przypisanych zadań

w czasie. Informacje te w postaci faktów kierowane są jako strumień wyjścia poprzez zastosowanie końcowych wyrażeń prezentujących dane.

Model DLP, który został stworzony na potrzeby przeprowadzenia badań przedstawiono w załączniku **A**.



## 4. Przeprowadzone badania

### 4.1 Cel i zakres badań

Celem badań jest określenie możliwości wykorzystania programowania techniką *Answer Set Programming* i ogólnodostępnych systemów ASP w rozwiązywaniu problemu harmonogramowania obserwacji satelitarnych. Ponadto, badania mają na celu określenie efektywności systemu *clingo* oraz sposobów wykorzystania jego możliwości na przykładzie tego problemu odnosząc się przy tym do innych, dotychczasowo stosowanych metod.

Zakres badań obejmuje przede wszystkim odpowiednie sformułowanie problemu obserwacji satelitarnych oraz stworzenie modelu DLP (ASP) na podstawie tego sformułowania umożliwiające analizę harmonogramowania dla konkretnych instancji problemu i otrzymanych wyników. Do przeprowadzenia badań konieczne jest również stworzenie generatora instancji (problemów), który w sposób zautomatyzowany i szybki może tworzyć instancje służące jako wejście dla systemu ASP i stworzonego modelu rozwiązania problemu. Dodatkowo w zakres prac wchodzi stworzenie algorytmu rozwiązującego problem harmonogramowania obserwacji satelitarnych sformułowany w identyczny sposób, do analizy porównawczej jego efektywności z efektywnością ASP.

### 4.2 Metodologia badań

W celu przeprowadzenia badań zastosowano następujący zestaw narzędzi:

#### 1. Zintegrowany system ASP – *clingo*

Wersja *clingo*:

- wersja 4.5.4
- wersja *gringo* 4.5.4
- konfiguracja: z Lua 5.2.3

Narzędzie *clingo*, stanowiące zintegrowane połączenie *groundera* z *solverem* wykorzystano jako podstawę do stworzenia modelu DLP, jest głównym narzędziem wykorzystanym do przeprowadzenia niniejszych badań. Stworzony model DLP (ASP) napisano w języku *gringo*, który jest interpretowalny dla zastosowanego systemu ASP i umożliwiający jego przetworzenie i wyszukanie rozwiązań w formie zbioru stabilnych modeli. Na potrzeby badań stworzono spójną definicję problemu rozdzielaając instancję problemu od modelu rozwiązania problemu na osobne pliki.

Podstawowe informacje dotyczące tego narzędzia umieszczono w rozdziale 3.1.<#>tutaj odnośnik</#>

2. Autorski zbiór narzędzi wspomagający wykonywanie badań posiadający funkcję generowania instancji problemu.

To narzędzie, napisane w języku *python* (ze względu na jego dużą popularność i szybkość w tworzeniu) można wykorzystać do wygenerowania losowych testowych instancji, dostosowując do potrzeb parametry takie jak np.:

- ilość orbit
- ilość zadań
- względny ilościowy rozkład zadań (na obserwacyjne, uplink, downlink)
- ilość zadań awaryjnych

Dodatkowo może ono służyć jako prosty parser dla wyjścia *clingo* (nakładka na *clingo*), ułatwiający pracę i analizę wyników pracy *solvera*. Podstawowe informacje w tym zakresie umieszczono w dodatku B [odnośnik](#)

3. Implementacja algorytmu *branch&bound* dla porównania modelu DLP z jedną ze standardowych metod optymalizacyjnych.

Algorytm napisano również w języku *python*, jest to prosta implementacja tej techniki optymalizacyjnej dla identycznego sformułowania problemu harmonogramowania, jak w przypadku systemu ASP. Ten algorytm jednak, zamiast z pliku instancji (deklaratywnego zapisu programu logicznego) korzysta z zapisu problemu w formacie JSON (*JavaScript Object Notation*), zawierającego dane dotyczące cech orbit i satelitów oraz zadań. Zapis ten jest w sensie logicznym tożsamy z zapisem, z którego korzysta system ASP, czyli zgodny z założeniami, które określono w rozdziale 3.1 [tutaj odnośnik](#). Jest to konieczne do utrzymania kryterium porównywalności tej metody ze stworzonym równolegle modelem DLP w badaniach.

Wszelkie prace prowadzone podczas niniejszych badań prowadzono na systemie linux w dystrybucji Linux Mint 20, 64-bit (jądro linux 4.15.0-20-generic), z dostępną pamięcią 7.7 GiB. Do uruchamiania kodu źródłowego będącego częścią niniejszej pracy również rekomenduje się używanie systemów z rodziny linux.

Jednym z celów prac badawczych jest przeprowadzenie analizy porównawczej, czyli porównanie działania i wyników systemu ASP na podstawie stworzonego modelu DLP ze standardowym algorytmem (w tym przypadku *branch&bound*) oraz analiza podobieństw lub różnic między tymi metodami. Porównywalność dla tych metod jest zachowana dzięki opracowaniu formalnego sformułowania problemu i niezbędnych założeń przy zastosowaniu ich w sposób jednolity podczas pisania kodu obsługującego obie metody i ich implementacje.

Na potrzeby badań założono następujące metryki:

- **czas trwania harmonogramowania:**

$$t = t_e - t_s$$

**gdzie:**

**t** – czas harmonogramowania [s]

**t<sub>s</sub>** – czas uruchomienia programu [s]

**t<sub>e</sub>** – czas zakończenia działania programu [s]

Czas działania to bardzo istotny element podczas badania problemów obliczeniowych. Dla problemów klasy złożoności problemów NP-trudnych, a do takich należy badany problem jest to szczególnie ważne, ponieważ dla tych problemów czas działania algorytmu rośnie bardzo szybko wraz z przyrostem rozmiaru danych wejściowych. W przypadku systemu *clingo* czas działania jest obliczany automatycznie i każdorazowo wyświetlany po zakończeniu działania programu (razem z dodatkowymi informacjami). W przypadku stworzonego algorytmu w wersji *branch&bound* czas działania również jest liczony automatycznie i wyświetlany po zakończeniu obliczeń. Czas działania oceniany jest z naciskiem na szybkość zakończenia obliczeń, czyli oczekiwana jest niska wartość czasu działania.

- **jakość harmonogramu**, określana jako:

$$q = \sum_{i=1}^n p_i \cdot x_i$$

**gdzie:**

**q** – jakość harmonogramu

**n** – całkowita ilość zadań obserwacyjnych

**p<sub>i</sub>** – priorytet zadania

**x<sub>i</sub>** – wartość binarna {0,1} opisująca stan przypisania zadania (1 jeśli zadanie przypisano do którejś z orbit, 0 jeśli zadania nie można wykonać lub pominięto je na rzecz przypisania zadań o wyższym priorytecie)

Jakość harmonogramu jest tym korzystniejsza im wyższa jest jej wartość. Obliczanie rozwiązania każdorazowo musi dążyć do zmaksymalizowania wartości jakości harmonogramu. W związku z tym staje się konieczne porównywanie i ocenianie zadań i ich priorytetów w przypadku, kiedy okna czasowe tych zadań nachodzą na siebie lub całkowicie się pokrywają, a jednocześnie ilość satelitów znajdujących się na orbitach, z których te zadania są widoczne jest zbyt mała aby wykonać je wszystkie. W tym przypadku może być konieczne pominięcie niektórych z tych zadań na rzecz zaplanowania wykonania innych, o wyższym priorytecie lub awaryjnych, których wykonanie jest obligatoryjne.

- **efektywność harmonogramowania**, określona jako:

$$e = \frac{q}{t}$$

**gdzie:**

**e** – efektywność harmonogramowania [1/s]

**q** – jakość harmonogramu

**t** – czas działania programu [s]

Efektywność harmonogramowania jest dodatkową wielkością pozwalającą na ewaluację szybkości postępu harmonogramowania, także dla każdej z zaimplementowanych metod rozwiązywania problemu. Wysoka efektywność oznacza, zdolność do znalezienia w krótkim czasie poprawnego harmonogramu o wyższej jakości. Wyższa efektywność oceniana jest zatem korzystniej, a niższa mniej korzystnie.

### **4.3 Charakterystyka badanych instancji problemu**

Do badań wykorzystano stworzony generator losowych instancji problemu. Ze względu na całkowitoliczbowy charakter większości danych opisujących problem i budujących daną instancję problemu taki generator nie wymaga skomplikowanych operacji generujących pseudolosowe wartości tych danych. Moduł ten korzysta tylko z natywnych funkcji języka python do uzyskania losowości w arbitralnie wyznaczonych dozwolonych ramach przy generowaniu danych takich jak:

- zasoby orbit (pamięć, energia akumulatora, generowanie energii)
- zużycie zasobów orbit (pamięć, energia)
- okna czasowe widoczności zadań
- typy poszczególnych zadań
- priorytety zadań.

Generator jest sparametryzowany. Oznacza to, że na jego wejściu (w pliku konfiguracyjnym) podaje się parametry dotyczące generowania danych dotyczących wygenerowanej instancji problemu. Dotyczy to min.

- docelowej ilości orbit w generowanej instancji
- docelowej ilości zadań w generowanej instancji

Więcej informacji dotyczących generatora losowych instancji problemu zamieszczono w dodatku A.

### **4.4 Wyniki badań**

<#>dokończyć, objaśnić (nawet to, co jest oczywiste)</#>

W celu analizy założonych metryk w pierwszym kroku zbadano wpływ ilości danych wejściowych (rozmiaru problemu) na działanie modelu. Wygenerowano łącznie 30 losowych instancji problemu, z których 10 ostatnich rozwiązano przy użyciu modelu DLP z systemem ASP. Następnie ponownie uruchomiono *clingo* rozszerzając jednak każdą z instancji o kolejne orbity, zwiększając tym samym ilość koniecznych do przeanalizowania kombinacji. W celu uzyskania bardziej ustabilizowanych i pewniejszych wyników, harmonogramowanie każdej z instancji w obu wersjach wykonano 5-cio krotnie z automatycznym obliczeniem średniej arytmetycznej czasu harmonogramowania, który przyjęto jako finalny czas działania  $t_i$ . Wpływ ilości (rozmiaru) danych w postaci analizowanych orbit na czas działania i jakość przypisania zadań przedstawiono w tabeli 1.

**Tabela 1.** Wpływ ilości analizowanych orbit na czas działania (harmonogramowania).

Instancja	2 orbity			4 orbity		
	t, s	q	e, 1/s	t, s	q	e, 1/s
21	3,468	334	96,309	6,631	358	53,989
22	0,086	316	3674,419	8,673	456	52,577
23	6,22	344	55,305	61,607	386	6,266
24	1,984	448	225,806	16,072	484	30,114
25	0,264	365	1382,576	8,218	419	50,986
26	0,064	0*	0*	8,448	435	51,491
27	0,146	182	1246,575	2,338	326	139,435
28	1,555	439	282,345	31,117	507	16,293
29	0,085	272	3200	1,611	315	195,531
30	2,451	357	145,655	15,071	405	26,873

0\* – akceptowalny (zgodny z ograniczeniami) harmonogram nie został znaleziony

Zmiana czasu trwania harmonogramowania przy zmianie rozmiaru problemu jest bardzo wyraźna, ale także bardzo zróżnicowana, gdyż czas potrzebny na dokonanie analizy problemu zależy od szczegółowych informacji zawartych w każdej z instancji (ilość zadań awaryjnych, zasoby satelitów, rozkład i czas trwania okien czasowych widoczności zadań, czas trwania wykonywania zadań przez satelity).

Dla wersji z 2 orbitami wartość czasu harmonogramowania instancji 21 - 30 mieści się w przedziale 64 ms do 6,22 s (kolejno instancje 26 i 23), dając średni czas harmonogramowania na poziomie **1,632 s**.

Najmniejszą wartość czasu harmonogramowania zanotowano w przypadku, w którym nie ma żadnego akceptowalnego rozwiązania problemu. System ASP zdołał zatem w bardzo krótkim czasie zakończyć harmonogramowanie zwracając komunikat o braku rozwiązań. Druga najmniejsza wartość czasu harmonogramowania odnosi się do instancji 29 z wynikiem 0,085 s i znalezionym optimum z jakością 272.

Jakość harmonogramu dla instancji 21 – 30 (z pominięciem 26, gdzie nie istnieje żaden poprawny harmonogram) mieści się w przedziale od 182 (instancja 27) do 448 (instancja 24). Średnia jakość harmonogramowania dla 2 orbit wynosi **305,8**.

Efektywność harmonogramowania przy 2 orbitach dla instancji 21 – 30 (z pominięciem 26, gdzie nie istnieje żaden poprawny harmonogram) mieści się w przedziale od 55,305 1/s (instancja 23) do 3674,419 1/s (instancja 22). Średnia efektywność harmonogramowania dla 2 orbit wynosi 1145,37.

Dla wersji rozszerzonej do 4 orbit wartość czasu harmonogramowania badanych instancji mieści się w przedziale 1,661 s do 61,607 s, (kolejno instancje 29 i 23) dając średni czas harmonogramowania na poziomie **15,979 s**. Dla instancji 26 przy 4 orbitach czas harmonogramowania wyniósł 8,448 s, po którym zakończono harmonogramowanie ze znalezionym optimum o jakości 435.

Jakość harmonogramu dla rozszerzonych instancji 21 – 30 mieści się w przedziale od 315 (instancja 29) do 507 (instancja 28). Średnia jakość harmonogramowania dla 4 orbit wynosi **409,1**.

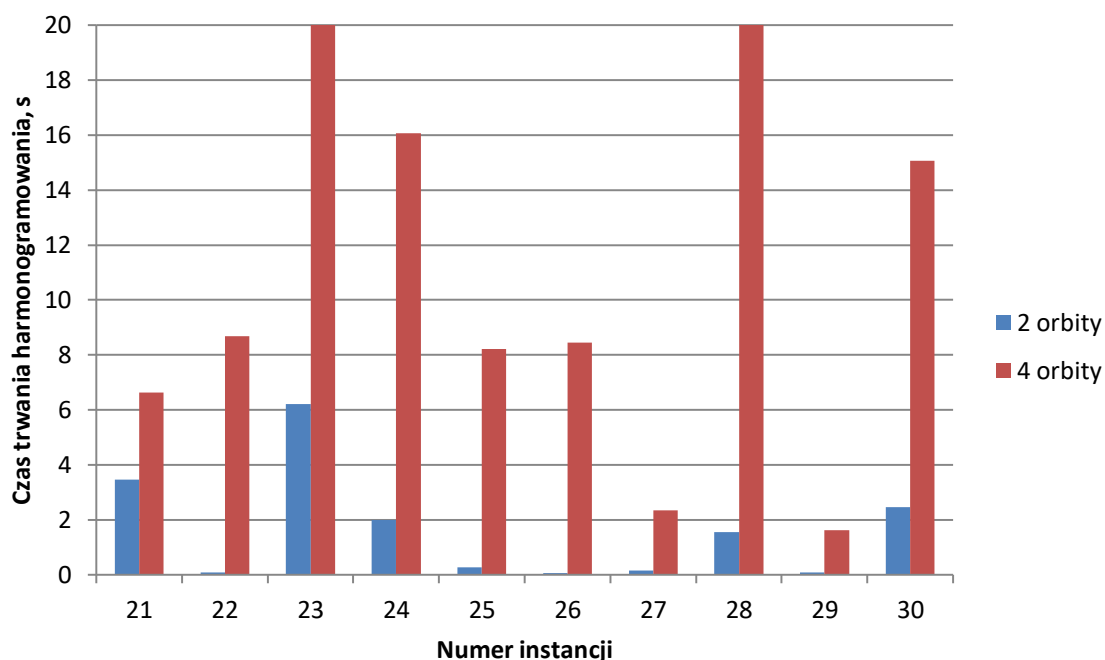
Efektywność harmonogramowania dla badanych instancji w wersji rozszerzonej mieści się w przedziale od 6,266 1/s (instancja 23) do 195,531 1/s (instancja 29). Średnia efektywność harmonogramowania dla 4 orbit wynosi **62,356 1/s**.

Przy zwiększeniu rozmiaru problemu w postaci ilości orbit zarówno czas harmonogramowania, jak i jakość harmonogramów znalezionych podczas analizy każdej z dotychczas zestawionych instancji uległy zmianie. Średni czas harmonogramowania wzrósł niemal 10-cio krotnie, przy dwukrotnym zwiększeniu ilości orbit. Polepszeniu uległa jednak średnia jakość harmonogramu – ok. 33%. Przy 4 orbitach znacznie wzrasta ilość możliwych kombinacji przypisania orbit do zadań, a co za tym idzie – ilość koniecznych do wykonania obliczeń. Jednak bezpośrednią korzyścią z większej ilości możliwych przypisań jest uzyskanie bardziej optymalnego przypisania orbit do zadań.

Przyrost czasu potrzebnego do przeprowadzenia harmonogramowania przy wzroście rozmiaru danych wejściowych oraz zróżnicowanie między poszczególnymi instancjami pokazano graficznie na wykresie 1.

**Wykres 1.** Czas trwania harmonogramowania w zależności od rozmiaru problemu.

## Czas trwania harmonogramowania w zależności od rozmiaru problemu (ilości orbit)



W kolejnym kroku przygotowano wszystkie 30 losowych instancji problemu do analizy porównawczej. Dokonano pomiarów czasu harmonogramowania dla każdej z instancji przy użyciu modelu DLP oraz równoległe metodą *branch&bound*. Zestawiono również obliczone wartości jakości harmonogramów oraz efektywności harmonogramowania. Dane dotyczące czasu działania, jakości oraz efektywności ukazano w tabeli 2.

**Tabela 2.** Porównanie czasu działania, jakości harmonogramu i efektywności harmonogramowania przy użyciu ASP oraz algorytmu *branch&bound*.

Instancja	ASP			b&b		
	t, s	q	e, 1/s	t,s	Q	e, 1/s
1	4,940	282	57,085	160,02	276	1,725
2	0,325	249	766,154	3,268	249	76,193
3	1,518	294	193,676	7,062	245	34,693
4	11,529	415	35,996	36,849	373	10,122
5	111,41	440	3,949	4,291	211	49,173
6	2,212	467	211,121	140,229	467	3,330
7	0,415	310	746,988	1031,515	310	0,301
8	5,462	352	64,445	61,199	352	5,752
9	2,049	462	225,476	153,138	462	3,017
10	0,157	314	2000	14,346	314	21,888
11	1,797	459	255,426	351,34	459	1,306
12	0,279	303	1086,022	162,804	303	1,861
13	0,668	361	540,419	51,994	361	6,943
14	6,444	402	62,384	56,933	402	7,061

15	0,044	0*	0*	2,882	0*	0*
16	0,21	309	1471,429	33,967	309	9,097
17	0,126	277	2198,413	56,898	277	4,868
18	0,988	410	414,980	34,971	410	11,724
19	0,498	344	690,763	27,137	344	12,676
20	0,847	349	412,043	139,872	349	2,495
21	3,207	334	104,147	359,958	334	0,928
22	0,069	316	4579,71	0,55	316	574,545
23	6,481	344	53,078	102,799	344	3,346
24	1,907	448	234,924	200,489	448	2,235
25	0,219	365	1666,667	4,324	365	84,413
26	0,061	0*	0*	3,169	0*	0*
27	0,133	182	1368,421	7,25	182	25,103
28	1,507	439	291,307	109,32	439	4,016
29	0,076	272	3578,947	20,214	272	13,456
30	2,296	357	155,488	169,081	357	2,111

0\* – akceptowalny (zgodny z ograniczeniami) harmonogram nie został znaleziony

### Harmonogramowanie z użyciem systemu *clingo*

Przeprowadzając harmonogramowanie techniką ASP dla instancji 1-30 wartość czasu harmonogramowania mieści się w przedziale 44 ms do 111,41 s (kolejno instancje 15 i 5), dając średni czas harmonogramowania na poziomie **5,596 s**.

Najmniejszą wartość czasu harmonogramowania zanotowano w przypadku instancji, przy których nie istnieje żaden akceptowalny harmonogram (instancje 15 i 26). System ASP w bardzo krótkim czasie zakończył harmonogramowanie zwracając komunikat o braku rozwiązań. Najlepszy czas działania przy jednoczesnej rozwiązywalności problemu to 69 ms (instancja 22).

Jakość harmonogramu dla instancji 1 – 30 (z pominięciem 15 i 26, dla których nie znaleziono rozwiązań) mieści się w przedziale od 182 (instancja 27) do 467 (instancja 6). Średnia jakość harmonogramowania dla tego zestawu instancji wynosi **352**.

Efektywność harmonogramowania dla badanych instancji (z pominięciem 15 i 26, dla których nie znaleziono rozwiązań) mieści się w przedziale od 3,949 1/s (instancja 5) do 4579,71 1/s (instancja 22). Średnia efektywność harmonogramowania dla tego zestawu instancji wynosi **838,195**.

### Harmonogramowanie techniką *branch&bound*

Przeprowadzając harmonogramowanie techniką *branch&bound* dla badanych instancji wartość czasu harmonogramowania mieści się w przedziale 550 ms do 1031,515 s (kolejno instancje 22 i 7), dając średni czas harmonogramowania na poziomie **116,929 s**.

Instancje, przy których nie istnieje żaden akceptowalny harmonogram (instancje 15 i 26) rozwiązano w krótkim czasie (kolejno 2,882 s i 3,169 s). Najlepszy czas działania - 550 ms



uzyskano tą metodą jednak dla instancji nr 22 przy jednoczesnym znalezieniu optimum (jakość 316).

Jakość harmonogramu dla instancji 1 – 30 (z pominięciem 15 i 26, dla których nie znaleziono rozwiązań) mieści się w przedziale od 182 (instancja 27) do 467 (instancja 6). Średnia jakość harmonogramowania *b&b* dla tego zestawu instancji wynosi **340,357**.

Efektywność harmonogramowania dla badanych instancji (z pominięciem 15 i 26, dla których nie znaleziono rozwiązań) mieści się w przedziale od 0,301 1/s (instancja 7) do 574,575 1/s (instancja 22). Średnia efektywność harmonogramowania dla tego zestawu instancji wynosi **34,8 1/s**.

Bardziej wnikliwa analiza porównawcza założonych metryk wymaga wyznaczenia np. następujących wielkości charakteryzujących podobieństwa i różnice w harmonogramowaniu między techniką ASP oraz metodą *branch&boud* dla analizowanych instancji:

1. Czas harmonogramowania porównano metodą względną, wyznaczając przyspieszenie harmonogramowania określone wzorem:

$$t' = \frac{t_{b\&b}}{t_{asp}}$$

**gdzie:**

$t'$  - przyspieszenie harmonogramowania,

$t_{b\&b}$  – czas działania dla instancji metodą *branch&bound* [s],

$t_{asp}$  - czas działania dla instancji metodą ASP [s],

Przyspieszenie harmonogramowania określa względne porównanie zoptymalizowania obliczeń podczas rozwiązywania problemu. Wartość przyspieszenia równa 1 oznacza brak przyspieszenia, a wartości większe od tej liczby oznaczają wielokrotne przyspieszenie obliczeń metodą ASP w porównaniu do *b&b*. Wartości mniejsze od 1 oznaczają, że wykonywanie harmonogramowania metodą ASP trwało dłużej niż metodą *b&b*.

2. Przyrost jakości harmonogramu określa polepszenie przypisania zadań do orbit. Oznacza to znalezienie bardziej optymalnego rozwiązania problemu. Wartość przyrostu większa od 0 oznacza zdolność systemu ASP do odnajdywania bardziej optymalnego harmonogramu niż metodą *b&b*. Przyrost jakości harmonogramu porównano metodą bezwzględną wg wzoru:

$$\Delta q = q_{asp} - q_{b\&b}$$

**gdzie:**

$\Delta q$  - przyrost jakości harmonogramu dla instancji,

$q_{b\&b}$  – jakość harmonogramu dla instancji metodą *branch&bound*,

$q_{asp}$  – jakość harmonogramu dla instancji metodą ASP,

Zestawienie przyspieszenia względnego oraz przyrostu jakości harmonogramu przedstawiono w tabeli 3.

**Tabela 3.** Analiza porównawcza harmonogramowania techniką ASP oraz harmonogramowania metodą *branch&bound*.

Instancja	$t'$	$\Delta q$
1	32,393	6
2	10,055	0
3	4,652	49
4	3,196	42
5	0,039	229
6	63,395	0
7	2485,578	0
8	11,205	0
9	74,738	0
10	91,376	0
11	195,515	0
12	583,527	0
13	77,835	0
14	8,835	0
15	65,500	0*
16	161,748	0
17	451,571	0
18	35,396	0
19	54,492	0
20	165,138	0
21	112,241	0
22	7,971	0
23	15,862	0
24	105,133	0
25	19,744	0
26	51,951	0*
27	54,511	0
28	72,541	0
29	265,974	0
30	73,642	0

0\* – akceptowalny (zgodny z ograniczeniami) harmonogram nie został znaleziony

Przeprowadzając harmonogramowanie dla zestawu instancji 1 – 30 przyspieszenie względne we wszystkich przypadkach oprócz instancji nr 5 osiągnęło wartość większą niż 1, co oznacza, że czas trwania harmonogramowania przy użyciu systemu ASP we wszystkich przypadkach poza instancją 5 jest wielokrotnie korzystniejszy niż dla metody *branch&bound*. Maksymalne przyspieszenie

uzyskano w przypadku nr 7 dla którego wartość wynosi 2485,578. Najmniejsza wartość przyspieszenia czasu harmonogramowania, a w zasadzie jego spowolnienie otrzymano dla instancji 5, przy której odnotowano wartość 0,039. Średnia wartość przyspieszenia harmonogramowania dla badanych przypadków wynosi **178,525**.

W większości badanych instancji uzyskana jakość harmonogramu uzyskana przy użyciu obu porównywanych metod była jednakowa. Odnotowano jednak przypadki, dla których system ASP zdołał odnaleźć bardziej optymalne rozwiązanie:

- dla instancji 1 uzyskano przyrost jakości harmonogramu o wartości 6 przy jednoczesnym przyspieszeniu harmonogramowania o wartości 32,393
- dla instancji 3 uzyskano przyrost jakości harmonogramu o wartości 49 przy jednoczesnym przyspieszeniu harmonogramowania o wartości 4,652
- dla instancji 4 wyniki są podobne jak dla instancji 3, z przyrostem jakości harmonogramu o wartości 42 i przyspieszeniem harmonogramowania o wartości 3,196
- dla instancji 5 doszło do zwolnienia czasu działania. Obliczenia z użyciem systemu ASP trwały w tym przypadku dłużej niż za pomocą metody b&b. Przyspieszenie o wartości 0,039 oznacza, że zaimplementowany algorytm *branch&bound* zakończył w tym przypadku obliczenia 25,641 razy szybciej, jednakże dla tego samego przypadku odnotowano przyrost jakości harmonogramu o wartości 229. W tym przypadku system ASP działał dłużej, ale zwrócił wynik o znacznie lepszej jakości. Ostateczne wyniki dla tej instancji to czas działania – 4,291 s. (b&b), 111,41 s. (ASP) oraz jakość rozwiązania - 211 (b&b), 440 (ASP).

Dla żadnej z badanych instancji algorytm *branch&bound* nie zdołał odnaleźć harmonogramu o lepszej jakości niż przy użyciu *clingo* (przyrost jakości harmonogramu nie jest ujemny dla żadnej z instancji).

## Wnioski

Wyniki czasu działania przy harmonogramowaniu z użyciem modelu DLP zwykle mieszczą się w 3 klasach czasowych

- poniżej 1 s – większość badanych instancji została rozwiązana właśnie w tym przedziale czasowym (instancje łatwo rozwiązywalne)
- poniżej 10 s – w tym czasie rozwiązano 10 instancji (instancje średnio trudne)
- więcej niż 10 – taka ilość czasu potrzebna jest na rozwiązanie 2 instancji (instancje trudno rozwiązywalne)

Przy zastosowaniu algorytmu optymalizacji *branch&bound* otrzymane wyniki czasu działania są bardziej zróżnicowane. Można natomiast zauważyć, że czas działania jest w stosunkowo dużej części badanych instancji o 1 lub 2 rzędy wielkości dłuższy niż czas rozwiązywania tych samych

instancji przy użyciu systemu ASP. Świadczą o tym wartości przyspieszenia harmonogramowania w przedziale [10;200] lub średni czas działania dla badanych instancji - **5,596 s.** (ASP) **116,929 s.** (b&b).

### Analiza trudności rozwiązywania instancji

System ASP jest w stanie zoptymalizować obliczenia podczas przeszukiwania przestrzeni rozwiązań. Narzędzie *clingo* ma możliwość wyświetlania statystyk dotyczących przebiegu działania programu. Przegląd tych statystyk w kontekście części badanych instancji przedstawiono w tabeli 4.

**Tabela 4.** Statystyki dotyczące ilości reguł, atomów, wyborów dotyczące poszczególnych instancji.

<dokończyć>

### Poszukiwanie rozwiązania o założonej jakości

Łatwa skalowalność w przypadku stosowania techniki ASP (a w zasadzie wynikająca z samego paradygmatu programowania w logice) pozwala min. na stawianie dodatkowych wymagań (reguł), które powinno spełniać poprawne rozwiązanie (poprawny harmonogram). W przypadku instancji o bardzo dużych rozmiarach znalezienie najlepszego możliwego harmonogramu z całej przestrzeni rozwiązań może zająć dużo czasu, jednak korzystając z cech charakterystycznych dla systemu ASP można postawić problem w nieco inny sposób, tj. zakończyć działanie programu po znalezieniu pierwszego rozwiązania spełniającego regułę jakości, otrzymując rozwiązanie, które w sposób szybki dostarcza akceptowalny wynik bez analizowania całej przestrzeni rozwiązań. Zobrazowano to w poniższej tabeli.

**Tabela 5.** Czas działania w przypadku poszukiwania pierwszego rozwiązania o wymaganej jakości (tzw. „do rozwiązania”).

Instancja	Najlepsze rozwiązanie (optimum)			Min 50% jakości			Min 60% jakości			Min 70% jakości			Maksymalna jakość (suma priorytetów)
	t, s	q	%	t, s	q	%	t, s	q	%	t, s	q	%	
31	75,226	778	77,96	0,439	551	55,21	0,442	601	60,22	0,492	728	72,95	998

Wyniki dotyczące czasu działania i jakości rozwiązania w przypadku instancji numer 31 oznaczają, że:

- maksymalna jakość, którą mógłby osiągnąć harmonogram z przypisania wszystkich zadań określonych w instancji problemu wynosi 998,
- maksymalna możliwa do uzyskania jakość harmonogramu to 778 (w tym konkretnym przypadku nie jest więc możliwe przypisanie wszystkich zadań)

- czas trwania harmonogramowania przy użyciu techniki ASP dla tej instancji w poszukiwaniu optimum (najlepszego możliwego do uzyskania harmonogramu) to ponad 75 sekund
- czas trwania harmonogramowania w przypadku oczekiwanej jakości minimum 50% wynosi jedynie 0,439 s. Czas ten rośnie nieznacznie wraz ze zwiększaniem oczekiwanej jakości do 70% w przypadku tej instancji

Tak postawiony problem wymaga uruchomienia modelu DLP rozszerzonego o dodatkową regułę (ograniczenie integralności), dodającą ograniczenie związane z akceptowalną jakością harmonogramu. W ten sposób można w pewien sposób zasymulować heurystyczne działanie modelu, przy czym kontroli podlega w tym przypadku minimalna akceptowalna jakość harmonogramu. Przeprowadzone dodatkowe pomiary na 5-ciu losowo wygenerowanych instancjach obrazują wyniki uzyskane przy rozszerzonym modelu DLP. Umieszczono je w poniższej tabeli:

**Tabela 6.** Zestawienie pomiarów czasu działania i jakości harmonogramu i efektywności rozszerzonego modelu DLP.

Instancja	$q_{\min}$ , %	t,s	q	e, 1/s
32	50	0,785	942	1200
	55	0,748	926	1237,968
	60	0,779	942	1209,243
	65	0,786	955	1215,013
	70	0,784	955	1218,112
	75	0,788	942	1195,431
	80	0,786	926	1178,117
	85	4,059	991	244,149
	87	3,697	1011	273,465
	89	3,72	1028	276,344
33	91	0*	0*	0*
	50	0,812	1152	1418,719
	55	0,742	1126	1517,520
	60	0,887	1047	1180,383
	65	0,873	1047	1199,313
	70	0,877	1047	1193,843
	75	0,884	1047	1184,389
	80	0,915	1047	1144,262
	85	9,028	1072	118,742
	87	0*	0*	0*
34	50	0,704	698	991,477
	55	0,708	698	985,876
	60	0,688	698	1014,535
	65	0,718	698	972,145
	70	0,709	698	984,485
	75	0,685	698	1018,978
	80	0,677	698	1031,019

Instancja	$q_{\min}$ , %	t,s	q	e, 1/s
	85	6,016	741	123,172
	87	3,83	788	205,744
	89	1,362	775	569,016
	91	4,447	791	177,873
	93	0*	0*	0*
35	50	0,802	610	760,599
	55	0,778	690	886,889
	60	0,773	733	948,254
	65	0,771	810	1050,584
	70	1,017	1073	1055,064
	75	0,935	980	1048,128
	80	0,941	980	1041,445
	85	1,219	1144	938,474
	87	1,347	1144	849,295
	89	1,573	1096	696,758
	91	1,371	1184	863,603
	93	1,273	1184	930,086
	95	1,312	1184	902,439
	97	0*	0*	0*
36	50	0,644	891	1383,540
	55	0,618	891	1441,748
	60	0,607	891	1467,875
	65	0,617	891	1444,084
	70	0,607	891	1467,875
	75	0,626	891	1423,323
	80	3,27	906	277,064
	85	7,072	969	137,019
	87	0*	0*	0*

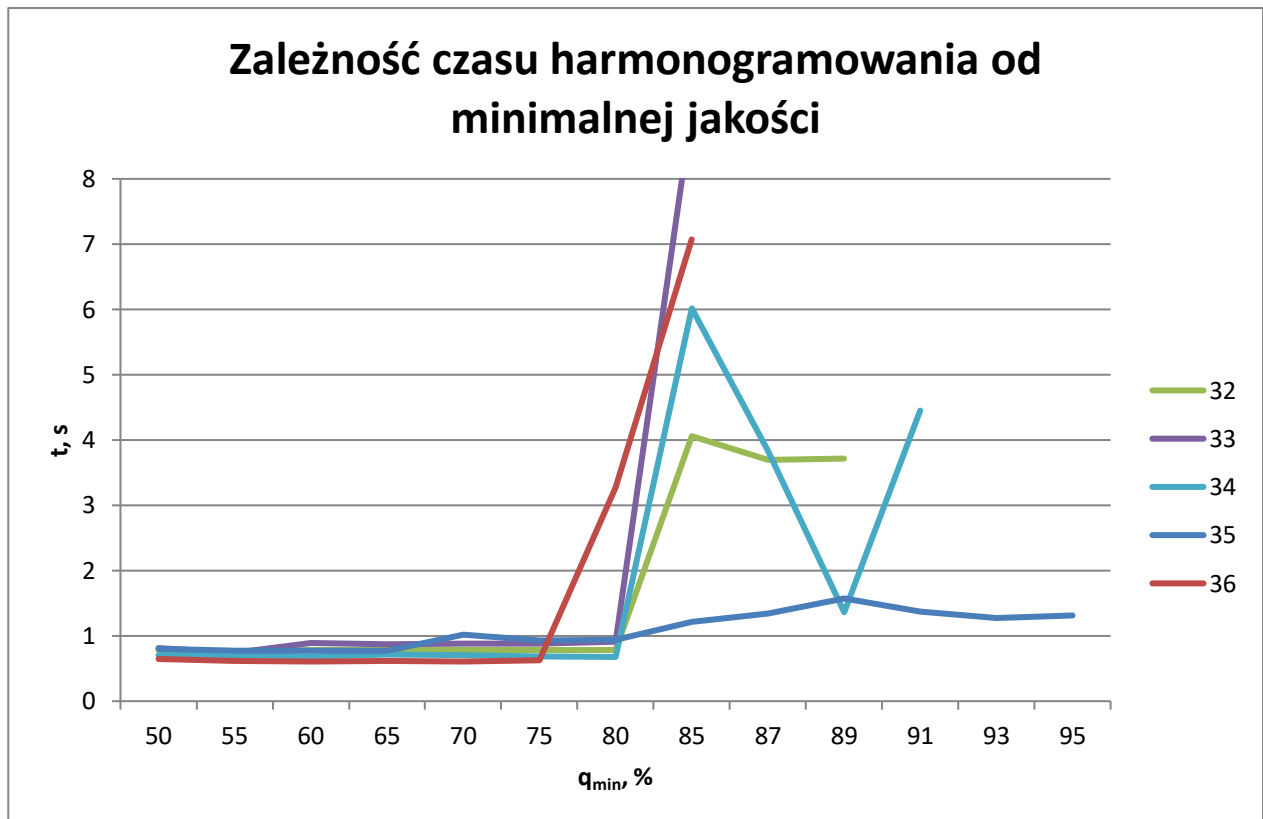
0\* – akceptowalny (zgodny z ograniczeniami) harmonogram nie został znaleziony

Przeprowadzając harmonogramowanie techniką ASP dla instancji 32-36 zastosowano rozszerzoną wersję modelu DLP w celu zbadania czasu harmonogramowania, uzyskanej jakości harmonogramu oraz efektywności harmonogramowania przy założonej minimalnej jakości. Instancje 32-36 to losowe problemy o wygenerowanej nieco większej złożoności (więcej orbit, zadań itp.). Wyniki działania rozszerzonego modelu DLP przedstawiają się następująco:

- przy niskiej wartości minimalnej założonej jakości (do ok. 80%) poprawny harmonogram znajdowany jest zawsze w czasie poniżej 1 sekundy. Dla wyższych wartości  $q_{\min}$ , czas ten zaczyna rosnąć gwałtownie w zależności od danej instancji aż do momentu, w którym system zmuszony jest przeanalizować wszystkie możliwe kombinacje, nie znajdując przy tym poprawnego harmonogramu (rozwiązanie o żądanej procentowej jakości nie istnieje w całej przestrzeni rozwiązań).
- jakość znalezionej harmonogramu rośnie wraz ze wzrostem  $q_{\min}$ , jednak nie jest to wzrost liniowy. Generalnie poprawny harmonogram jest znajdowany bardzo łatwo, a jego jakość zwykle

znacznie przekracza próg narzucany przez  $q_{\min}$ . Dla wyższej wymaganej jakości (w zależności od instancji) jakość harmonogramu zaczyna rosnąć już proporcjonalnie do  $q_{\min}$ .

- efektywność harmonogramowania dla badanych rozszerzonych instancji jest odzwierciedleniem czasu harmonogramowania, który narasta bardzo szybko dla wysokich wartości  $q_{\min}$ . Efektywność w tym przypadku skokowo spada, ze względu na to, że stosunkowo niewielki uzysk jakości harmonogramu (bardziej optymalne rozwiązanie) odbywa się kosztem bardzo dużego przyrostu czasu potrzebnego na wykonanie obliczeń.



## 5. Podsumowanie

### BIBLIOGRAFIA

**<#>zrobić porządnie</#>**

*[a] Materiały i zbiory danych dostępna na stronie <https://www.ucsusa.org> (csv)*

**[1] – Słownik języka polskiego <https://sjp.pl/>**

**[2] - encyklopedia britanica <https://www.britannica.com/science/orbit-astronomy>**

**[3] - Autonomiczny wieloczujnikowy system satelitarny na wieloagentowym modelu tablicowym**

**[4] - Satellite Communications Systems Engineering Atmospheric Effects, Satellite Link Designand System Performance**

**[5] Clingo guide – potassco**

**[6] Introduction to the theory of computation**

**[8] Legal and Political Aspects of the Use of European Satellite Navigation Systems Galileo and EGNOS**

**[9] SATELITARNE sieci teleinformatyczne**

**[10] Bellis, Mary. "The History of Satellites - Sputnik I." ThoughtCo. <https://www.thoughtco.com/history-of-satellites-4070932> (dostęp 28.08.2020).**

**[11] Keith Schmidt, 2001, Using Tabu Search to Solve the Job Shop Scheduling Problem with Sequence Dependent Setup Times**

**[12] Programista 3/2018 (70) 285358**

**[13] AU-18 Space Primer, Prepared by Air Command And Staff College Space Research Electives Seminars, Air University Press Maxwell Air Force Base, Alabama**

**[14] Prolog**

**[15] Applications of Answer Set Programming Esra Erdem, Michael Gelfond, Nicola Leone, 2016, Association for the Advancement of Artificial Intelligence. ISSN 0738-4602**



**[16] A SIMPLE OPTIMISED SEARCH HEURISTIC FOR THE JOB-SHOP SCHEDULING PROBLEM**

**[17] Introduction to logic programming**

**[18] Goodman, Melissa D.; Dowsland, Kathryn A.; Thompson, Jonathan M. (2007). "A grasp-knapsack hybrid for a nurse-scheduling problem"**

**[19] A Constraint-Based Approach to Satellite Scheduling Joseph C. Pemberton and Flavius Galiber, Pacific-Sierra Research 1400 Key Boulevard, Suite 700 Arlington, VA 22209**

**[20] A SIMPLE OPTIMISED SEARCH HEURISTIC FOR THE JOB-SHOP SCHEDULING PROBLEM**

**[21] Satellite observation scheduling with a novel adaptive simulated annealing algorithm and a dynamic task clustering strategy**

**[22] Earth observation satellite scheduling for emergency tasks**

**[23] GrinGo: A New Grounder for Answer Set Programming**

**[24] Answer Set Programming: Language, Applications and Development Tools**

**[25] Potassco user guide - potassco**