

Dokumentacja projektu “Planowanie farmy”

Kacper Rogalski
Hubert Wilga
Mateusz Tężyński

Opis problemu	3
Ogólny opis	3
Model	4
MIP start	5
Instrukcja	6
Github	6
Uruchomienie przez CPLEX	6
Uruchomienie przez Visual Studio z MIP	8
Analiza wyników	12

Opis problemu

Ogólny opis

Ten problem dotyczy planowania produkcji na farmie o powierzchni 200 akrów przez następne pięć lat, opierając się na pracy Swarta i współpracowników z 1975 roku. Obecnie rolnik posiada stado 120 krów, w tym 20 jałówek i 100 krów mlecznych. Zarządzanie stadem wymaga uwzględnienia różnych aspektów, takich jak potrzeba pastwisk dla krów, cieląt, sprzedaż/rozrastanie cieląt, utrata krów i jałówek oraz rotacja krów mlecznych.

Każda jałówka wymaga $\frac{2}{3}$ akra pastwiska, a każda krowa mleczna – 1 akra. Krowy mleczne rodzą średnio 1,1 cielaka rocznie, z czego połowa to byczki sprzedawane natychmiast za 30 funtów każdy. Pozostałe jałówki można sprzedać natychmiast za 40 funtów lub hodować przez dwa lata, aby stały się krowami mlecznymi. Krowy mleczne są sprzedawane w wieku 12 lat za 120 funtów każda. Należy jednak uwzględnić roczną stratę 5% jałówek i 2% krów mlecznych. Aktualnie w stadzie znajduje się 10 krów w wieku od noworodków do 11 lat.

W kontekście modelu na pięć lat, wprowadza się dużą liczbę zmiennych reprezentujących liczby krów w różnym wieku w każdym roku. Wartości tych zmiennych są ustalane przez początkowe dane oraz przez ograniczenia modelu. Choć obliczenie tych wartości ręcznie mogłoby sprawić, że model byłby bardziej kompaktowy, jego zrozumienie byłoby trudniejsze. Dlatego warto skorzystać z komputera, który przeprowadzi obliczenia, aby stworzyć bardziej przejrzysty i zrozumiały model.

Model

```
int MaxAge = ...;
range AgeNbs = 1..MaxAge;

range CowAges = 2..MaxAge-1;

float InitialCows[AgeNbs] = ...;
float HeifSurvival = ...;
float CowSurvival = ...;
float CalfRate = ...;
float HeifFraction = ...;
float InitialCap = ...;
float GrainPerCow = ...;
float SugarbeetPerCow = ...;
float GrainPerAcre[LandNbs] = ...;
float GrainAcre[LandNbs] = ...;
float SugarbeetPerAcre = ...;
float HeifAcre = ...;
float CowAcre = ...;
float Acres = ...;
float HeifLabor = ...;
float CowLabor = ...;
float Labor = ...;
float GrainLabor = ...;
float SugarbeetLabor = ...;
float DownCowChange = ...;
float UpCowChange = ...;
float BullockSellPrice = ...;
float HeifSellPrice = ...;
float CowSellPrice = ...;
float MilkPerCowPrice = ...;
float GrainSellPrice = ...;
float SugarbeetSellPrice = ...;
float GrainBuyPrice = ...;
float SugarbeetBuyPrice = ...;
float LaborFixCost = ...;
float LaborCost = ...;
float HeifCost = ...;
float CowCost = ...;
float GrainGrowCost = ...;
float SugarbeetGrowCost = ...;
int LoanTerm = ...;
float Repay = ...;
```

Zmienne:

Na powyższym obrazie widać wszystkie zmienne wykorzystane aby zbudować nasz model, większość zmiennych jest typu float, aby dać sobie możliwość, bardzo szczegółowego ustawienia prawdopodobieństw/wzrostów/spadków/szans. W naszym modelu możemy również ustawić zasięg(range) które pozwalają na łatwe iterowanie np, przez wszystkie grupy wiekowe krów.

Ograniczenia:

Ograniczenia modelu CPLEX uwzględniają różne aspekty zarządzania stadem i zasobami gospodarstwa. Ustalają liczbę krów na każdy rok, uwzględniając przeżywalność i starzenie się krów, aby zapewnić zrównoważony rozwój stada. Dodatkowo, ograniczenia dotyczące powierzchni, zasobów zboża i buraków, a także

pracy, zapewniają, że gospodarstwo funkcjonuje w granicach dostępnych zasobów i ponoszonych kosztów, przy jednoczesnym maksymalizowaniu zysków i uwzględnieniu zobowiązań finansowych.

MIP start

W naszym projekcie, po zaimplementowaniu modelu CPLEX w C++, zdecydowaliśmy się na wykorzystanie techniki MIP start, aby zoptymalizować proces znajdowania rozwiązań. MIP start pozwala na wprowadzenie początkowego rozwiązania do modelu optymalizacyjnego CPLEX.

Korzystanie z MIP start ma kilka kluczowych zalet. Po pierwsze, dostarczenie początkowego rozwiązania może znacząco przyspieszyć proces konwergencji do optymalnego rozwiązania. Zamiast rozpoczynać optymalizację od zera, CPLEX może zacząć od już częściowo rozwiązanego modelu, co redukuje liczbę iteracji potrzebnych do osiągnięcia optymalności.

Po drugie, MIP start pomaga w uniknięciu problemów z rozwiązaniami lokalnymi i może prowadzić do lepszego wykorzystania dostępnych zasobów obliczeniowych. Poprzez wprowadzenie dobrze zaprojektowanego początkowego rozwiązania, można również zmniejszyć ryzyko utknięcia w suboptymalnych punktach, co jest szczególnie istotne w przypadku złożonych modeli.

W praktyce, implementacja MIP start w naszym modelu CPLEX w C++ polegała na obliczeniu i dostarczeniu zestawu wartości początkowych dla zmiennych decyzyjnych modelu. Te wartości zostały następnie wprowadzone do CPLEX-a przed rozpoczęciem właściwej optymalizacji. Dzięki temu byliśmy w stanie skrócić czas potrzebny na znalezienie rozwiązania, zwiększając jednocześnie efektywność i dokładność procesu optymalizacji.

W rezultacie, implementacja MIP start nie tylko przyspieszyła proces optymalizacji, ale również poprawiła ogólną jakość uzyskiwanych wyników, co jest kluczowe dla skutecznego zarządzania zasobami w naszym modelu rolniczym.

Warto także wspomnieć, że w celu dostarczenia danych wejściowych do modelu CPLEX, wykorzystaliśmy dwa pliki .dat, które zawierały niezbędne informacje dotyczące parametrów modelu. Dzięki temu mieliśmy możliwość łatwego dostosowania danych wejściowych bez konieczności ingerencji w kod programu. To podejście umożliwiło nam skuteczne zarządzanie danymi i zapewniło elastyczność w eksperymentowaniu z różnymi konfiguracjami modelu.

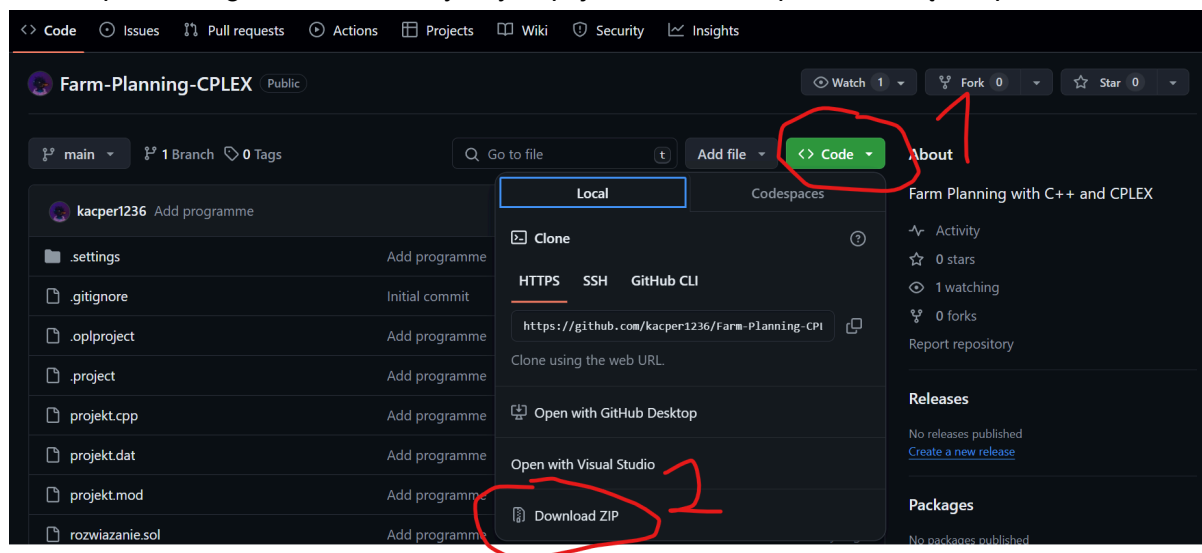
Instrukcja

Github

Projekt został udostępniony na platformie GitHub, pod adresem url:

<https://github.com/kacper1236/Farm-Planning-CPLEX>.

W celu pobrania go lokalnie należy użyć opcji “klonowania” przez stronę lub przez terminal.



W przypadku użycia strony należy użyć zaznaczonych przycisków w danej kolejności, pobrane zostanie archiwum, które należy rozpakować do wybranego folderu.

Jeżeli jednak wybierzemy opcję klonowanie przez terminal, należy przejść w nim do wybranej ścieżki, a następnie użyć polecenia:

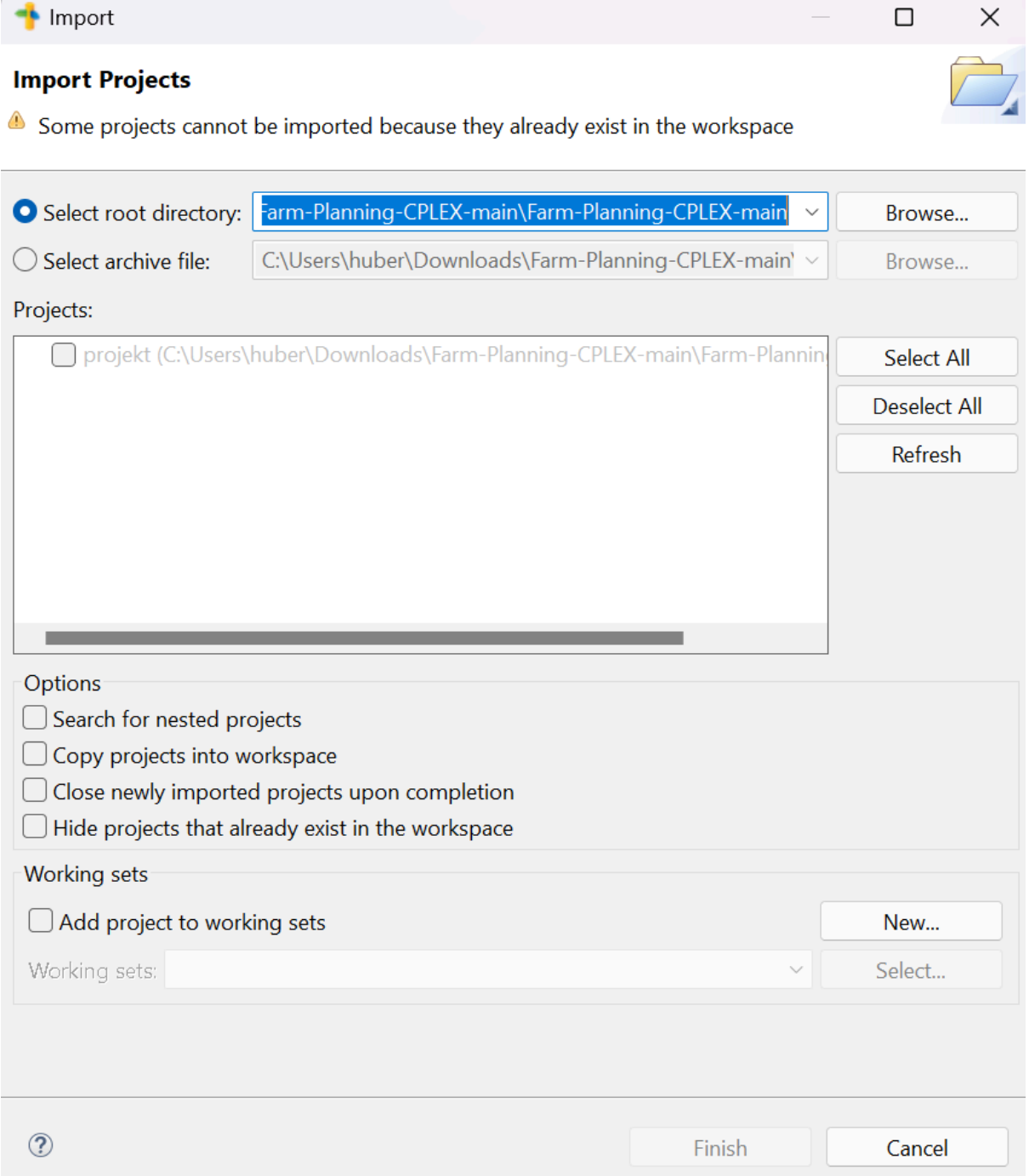
```
git clone https://github.com/kacper1236/Farm-Planning-CPLEX.git
```

Nie ważne od wybranego sposobu pobrania repozytorium, po tym przygotowaniu można przejść do kolejnych kroków.

Uruchomienie przez CPLEX

1. Należy uruchomić CPLEX Studio IDE (projekt był tworzony w wersji 22.1.1)
2. Wybrać kolejno “File>Import>Existing OPL projects”

3. W poniższym oknie wybieramy "Select root directory" i po kliknięciu "Browse" wybieramy folder w którym znajdują się wcześniej pobrane pliki projektu



Import

Import Projects

Some projects cannot be imported because they already exist in the workspace

☒ Select root directory: Farm-Planning-CPLEX-main\Farm-Planning-CPLEX-main

☐ Select archive file: C:\Users\huber\Downloads\Farm-Planning-CPLEX-main\

Projects:

☐ projekt (C:\Users\huber\Downloads\Farm-Planning-CPLEX-main\Farm-Planning-CP...

Options

☐ Search for nested projects

☐ Copy projects into workspace

☐ Close newly imported projects upon completion

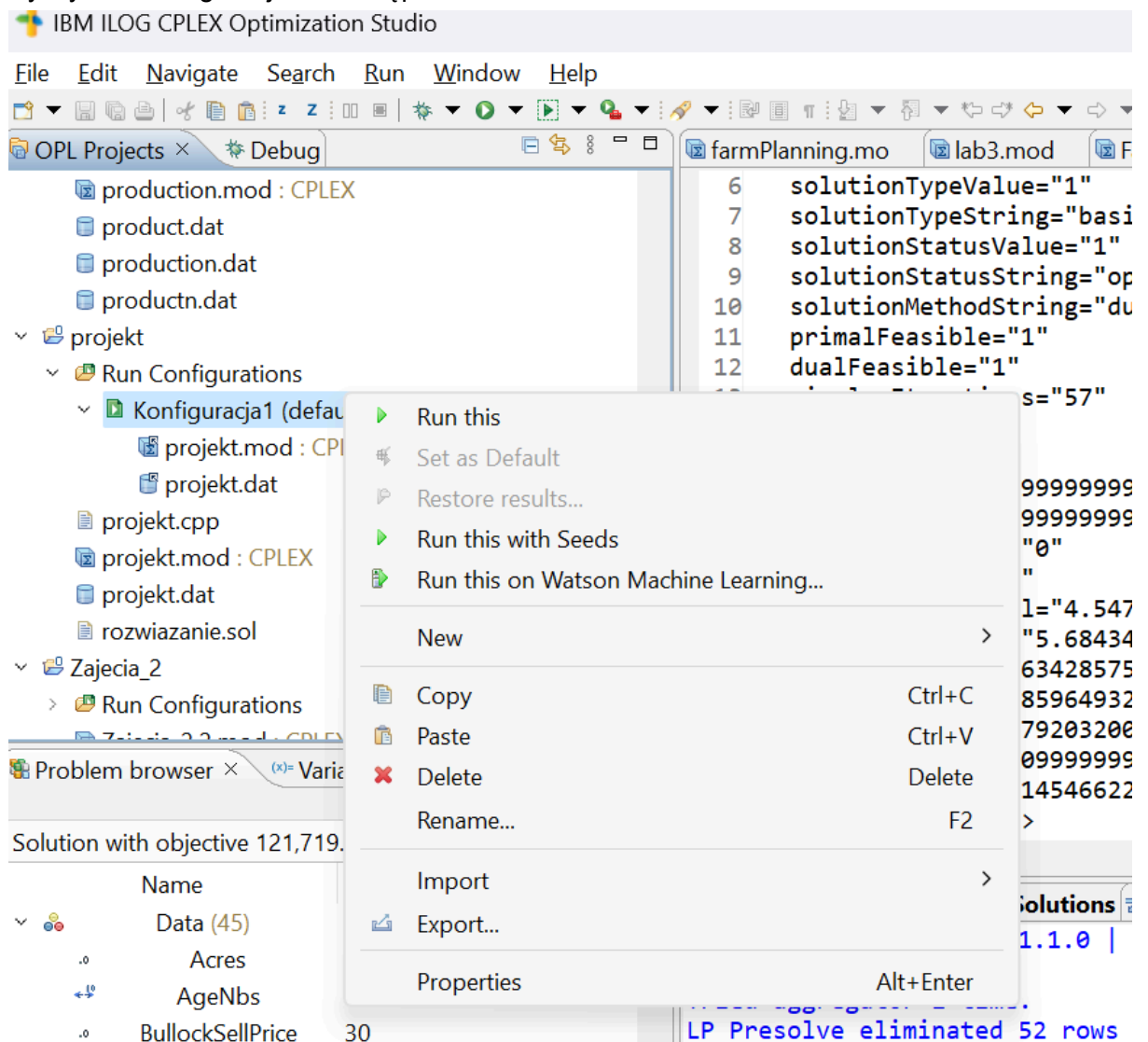
☐ Hide projects that already exist in the workspace

Working sets

☐ Add project to working sets

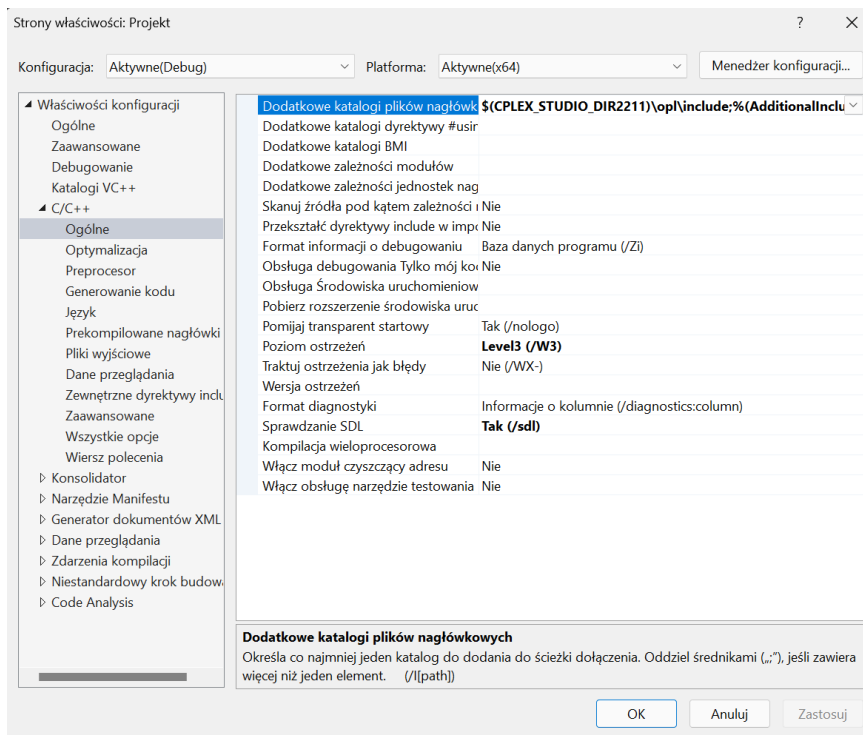
Working sets:

- Projekt jest gotowy do uruchomienia, aby to zrobić klikamy prawym przyciskiem myszy na "Konfiguracja1" i następnie "Run this"

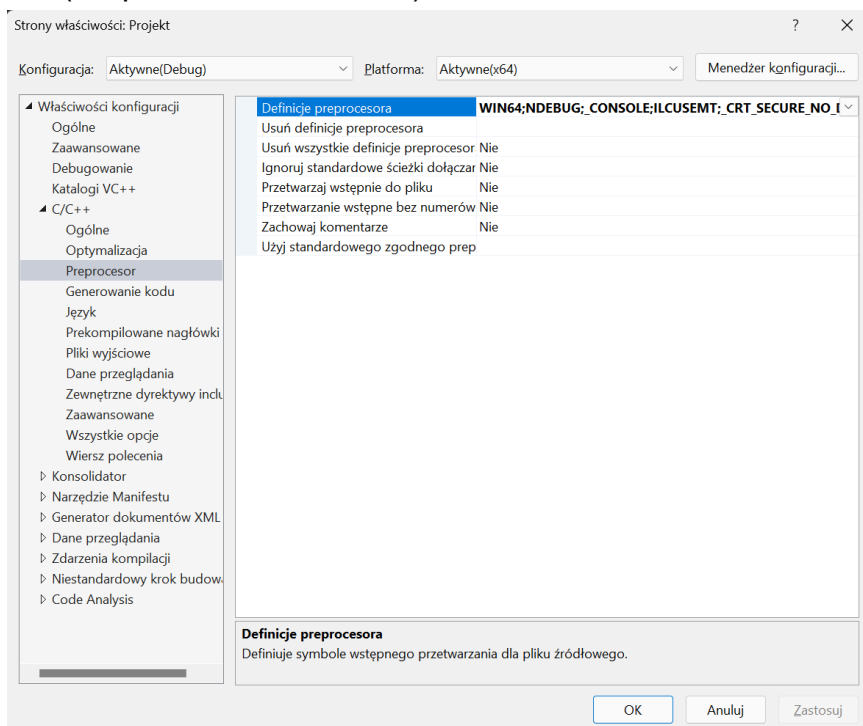


Uruchomienie przez Visual Studio z MIP

- Uruchamiamy Visual Studio i wybieramy folder pobrany i ewentualnie rozpakowany z GitHub.
- Wybieramy z górnego menu "Projekt>Projekt właściwości", wybieramy "Właściwości konfiguracji>C/C++>Ogólne" i zmieniamy zaznaczoną wartość na ``$(CPLEX_STUDIO_DIR2211)\opl\include;%(AdditionalIncludeDirectories)``

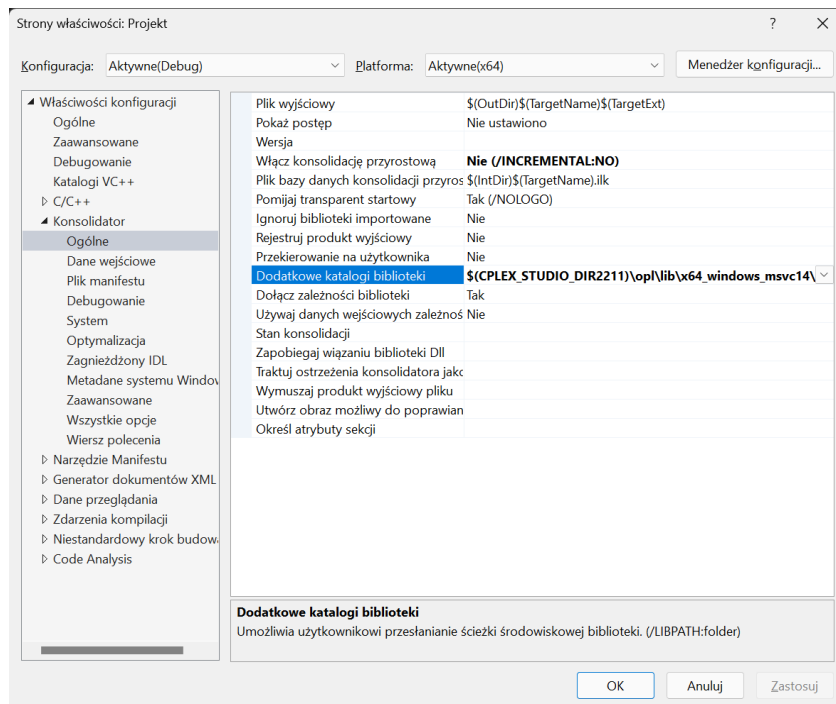


3. W tym samym menu zmieniamy zakładkę na "Preprocesor" i ponownie zaznaczoną wartością zmieniamy na ``WIN64;NDEBUG;_CONSOLE;ILCUSEMT;_CRT_SECURE_NO_DEPRECATED;%(PreprocessorDefinitions)``

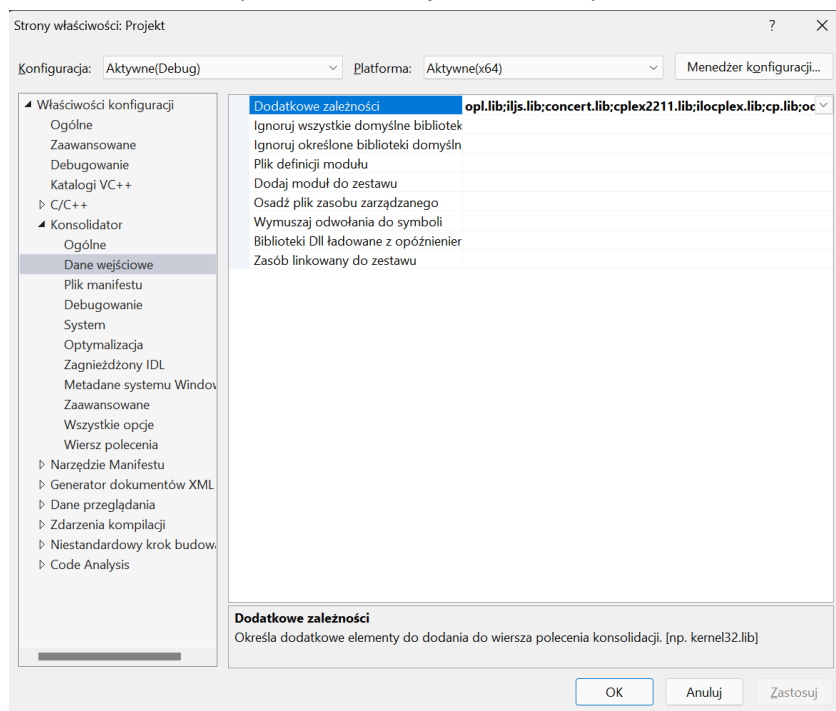


4. Zmieniamy zakładkę na "Konsolidator>Ogólne" i podświetlaną opcję zmieniamy na ``$(CPLX_STUDIO_DIR2211)\op\lib\x64_windows_msvc14\stat_mda;%(Add``

itionalLibraryDirectories)`



5. W zakładce "Dane wejściowe" zmieniamy ponownie podświetlaną poniżej wartość na:
``opl.lib;iljs.lib;concert.lib;cplex2211.lib;ilocplex.lib;cp.lib;odbc32.lib;odbc32.lib;wsock32.lib;%(AdditionalDependencies)``



6. Zamykamy to okno i otwieramy plik `projekt.cpp`, a w nim zmieniamy linijki z komentarzem `//Need to edit`. Powinny się tam znaleźć bezwzględne ścieżki do zadanych plików modeli i skryptów opl.
7. Z górnego menu wybieramy przycisk z zieloną strzałką `Lokalny debugger Windows`, co uruchamia projekt.

8. W nowo otwartym oknie wiersza poleceń powinny być widoczne wyniki, najpierw pierwsze, a następnie po MIP start.

Analiza wyników

Optymalizacja:

- CPLEX użyty do rozwiązania problemu optymalizacji.
- Wstępne przekształcenia (presolve) zmniejszyły liczbę wierszy i kolumn w modelu.
- Proces iteracyjny zakończył się osiągnięciem optymalnego rozwiązania po 47 iteracjach.

Wynik:

- Obiektów: 121719.17 - Jest to wartość funkcji celu, którą maksymalizujemy (najprawdopodobniej zysk).
- Stan: Rozwiązanie optymalne.
- Jakość: Brak niezgodności ograniczeń i kosztów zredukowanych.
- Gospodarstwo i Produkcja:

Zyski (Profit):

- Zyski z poszczególnych lat to: 21906, 21889, 25816, 26826, 25283.

Całkowita liczba krów (CowTotal):

- W kolejnych latach: 97.7, 95.167, 104.43, 103.57, 92.461.

Liczba krów (Cow):

Macierz pokazuje rozkład liczby krów w poszczególnych grupach w ciągu lat. Przykładowo, w pierwszym roku mamy 9.5 krów w pierwszej grupie, 21.66 w drugiej, 11.005 w trzeciej, 0 w czwartej i piątej grupie.

Sprzedaż małych krów (SmallCow):

Sprzedaż w pierwszym roku wyniosła 22.8, w drugim 11.584, a w kolejnych latach brak sprzedaży.

Sprzedaż HeifSell(Małych krów):

Sprzedaż w poszczególnych latach wynosiła: 30.935, 40.757, 57.436, 56.964, 50.853.

Uprawa zboża (GrainGrow):

Uprawy zboża w hektarach: w pierwszym roku na każdym polu 22 hektary, w drugim roku tylko na jednym polu 2.8207 hektara, w pozostałych latach brak upraw.

Zakup zboża (GrowBuy):

Zakup zboża w tonach: 36.62, 35.1, 37.837, 40.143, 33.476.

Sprzedaż zboża (GrainSell):

Brak sprzedaży zboża we wszystkich latach.

Uprawa buraków cukrowych (SugarbeetGrow):

Uprawy buraków cukrowych w hektarach: 91.15, 94.005, 97.65, 114.64, 131.31.

Zakup buraków cukrowych (SugarbeetBuy):

Brak zakupu buraków cukrowych we wszystkich latach.

Sprzedaż buraków cukrowych (SugarbeetSell):

Sprzedaż buraków cukrowych w tonach: 22.76, 27.388, 24.55, 42.143, 66.586.