Radosław Pytlak, Tomasz Zawadzki, Damian Suski
Wydział Mechatroniki, Politechnika Warszawska

Wojciech Stecz
Wojskowa Akademia Techniczna, Wydział Cybernetyki

System wspomagania pracy służb sanitarnych

Support System of sanitary teams work

W artykule opisany został system zarządzania służbami sanitarnymi stacji sanitarno-epidemiologicznych. System wspomaga pracę analityków i ma zastosowanie w przypadku wystąpienia ryzyka wybuchu epidemii. Pomyślano go tak, aby dostarczyć narzędzi pozwalających ocenić rozmiar epidemii na podstawie aktualnie posiadanych danych. Ponieważ służby sanitarne starają się kontrolować rozprzestrzenianie się epidemii, zaprezentowane narzędzie uwzględnia model działania takich służb. Wynikiem modelu jest opracowanie zbioru równań różniczkowych opisujących aktywność służb sanitarnych. Artykuł opisuje szczegółowo wybrane elementy systemu. W szczególności zwrócono uwagę na komponent do budowy modelu formalnego (opisującego nie tylko model epidemii, ale także działania samych służb). Zaprezentowano sposób kalibracji modelu. Przedstawione rozwiązanie jest częścią pakietu dostępnego z poziomu sieci Internet. Na zakończenie pokazano sposób konstrukcji zadania optymalizacji i jego rozwiązania.

Słowa kluczowe:
modele epidemiczne, modele Forrestera aktywności służb sanitarnych, kalibracja modeli.

We describe the information system that has been built for the support sanitary teams. The system is aimed at supporting analytical work which must be carried out when there is a risk of epidemic outbreak. It is meant to provide tools for predicting the size of an epidemic on the basis of the actual data collected during its course. Since sanitary teams try to control the size of the epidemics such a tool must model also sanitary teams activities. As a result a model for the prediction can be quite complicated in terms of the number of equations it contains. Furthermore, since a model is based on several parameters there must be a tool for finding these parameters on the basis on the actual data corresponding to the epidemic evolution. The paper describes the proposition of such a system. It presents, in some details, the main components of the system. In particular, the environment for building complex models (containing not only the epidemic model but also activities of sanitary teams trying to inhibit the epidemic) is discussed. Then, the module for a model calibration is presented. The module is a part of server for solving optimal control problems and can be accessed via Internet. Finally, we show how optimal control problems can be constructed with the aim of the efficient epidemic management. Some optimal control problems related to that issue are discussed and numerical results of its solution are presented.

Key words:
epidemic models, Forrester's models of sanitary activities, model calibration.

Wprowadzenie

W artykule opisany został system zarządzania służbami sanitarnymi stacji sanitarno-epidemiologicznych. System wspomaga pracę analityków i ma zastosowanie w przypadku wystąpienia ryzyka wybuchu epidemii oraz w przypadku wybuchu epidemii. Pomyślano go tak, aby dostarczyć narzędzi pozwalających ocenić rozmiar epidemii na podstawie aktualnie posiadanych danych. Ponieważ służby sanitarne kontrolują stan epidemii na obszarze, model uwzględniać musi także działania takich służb w rejonie. W rezultacie model składa się z wielu równań opisujących działania służb i rozwój epidemii jako takiej. Ponieważ sam model zawiera wiele parametrów, musieliśmy opracować narzędzie dla potrzeb określania wartości tychże parametrów. Artykuł opisuje szczegółowo wybrane elementy systemu. W szczególności zwrócono uwagę na komponent do budowy modelu formalnego (opisującego nie tylko model epidemii, ale także działania samych służb). Zaprezentowano sposób kalibracji modelu. Przedstawione rozwiązanie jest częścią pakietu dostępnego z poziomu sieci Internet. Na zakończenie pokazano sposób konstrukcji zadania optymalizacji i jego rozwiązania. Omówione zostały także wyniki przykładowych zadań optymalizacji.

Środowisko do modelowania procesu działania służb sanitarnych w zakresie zwalczania epidemii składa się z kilku głównych komponentów. Są to baza danych, podsystem modelowania i symulacji oraz podsystem optymalizacji. Główny komponent systemu modeluje rozwój epidemii. Co więcej, w przeciwieństwie do dotychczasowych prac z literatury światowej zakładamy, że modelowana jest także aktywność służb sanitarnych, które mają bezpośredni wpływ na rozwój epidemii. Komponent odpowiedzialny za wsparcie modelowania opisany jest na rysunku jako JOptisim Simulator. Komponent posiada graficzny interfejs do modelowania złożonych systemów — zasady modelowania podane zostaną pokrótce w dalszej części artykułu.

Kolejnym komponentem jest podsystem odpowiedzialny za optymalizację modeli opisanych w JOptisim Simulator. Podsystem wykonuje następujące zadania: kalibracja modelu i optymalizacja działań służb sanitarnych nastawiona na powstrzymanie rozwoju epidemii. Oprócz powyższych komponentów występuje jeszcze podsystem odpowiedzialny za dostarczanie danych do modelu. Podsystem określony jako DB — jest magazynem przechowującym informacje na temat liczby aktualnych zachorowań, rodzaju aktualnie wykonywanych działań itd. Przechowuje on także całość danych historycznych na powyższe tematy. Baza danych zawiera informacje o zakażonej żywności i jej spożyciu, co jest szczególnie przydatne do modelowania zakażeń rozchodzących się drogą pokarmową.

Inne komponenty to DOML File Generator oraz Optimizer GUI. Ten pierwszy ma za zadanie transformację modelu na język zrozumiały dla optymalizatora. W praktyce oznacza to zapisanie równań różniczkowo-algebraicznych do postaci modelu opartego jedynie o równania różniczkowe zwyczajne. Optimizer GUI wspiera definiowanie samego zadania optymalizacji i metody rozwiązania.

Komponent do modelowania procesów

Komponent do modelowania procesów wykorzystuje metodykę Forrestera (Sterman, 2000) do budowy modeli dynamicznych. Metodyka została zaproponowana przez Forrestera w latach pięćdziesiątych ubiegłego wieku i początkowo była wykorzystywana do modelowania zachowania organizacji biznesowych — głównie w przemyśle. W późniejszym okresie zaczęto ją wykorzystywać do innych zadań, w tym do modelowania procesów decyzyjnych. Zgodnie z koncepcją Forrestera model powstaje w dwóch etapach. W pierwszym etapie wybierane są zmienne modelu oraz określane są zależności między nimi. Powstaje wówczas diagram CLD, który odwzorowuje sprzężenia zwrotne między poszczególnymi zmiennymi. Używa się go do weryfikacji poprawności przygotowanego modelu. Przykładowo, jeżeli wiemy z danych historycznych, że wartość jednej ze zmiennych wzrasta wykładniczo, wówczas musi występować w modelu przynajmniej jedno sprzężenie zwrotne dodatnie związane z tą zmienną. Na drugim etapie budowy modelu generowane są równania różniczkowo-algebraiczne. Oznacza to, że zmienne modelu generują zmienne różniczkowe i związane z nimi równania określające zmiany wartości zmiennych w czasie. Do tych równań przypisane są pozostałe parametry modelu. Zazwyczaj równania różniczkowe są nieliniowe, co pokazuje złożoność procesu modelowania.

Przykład

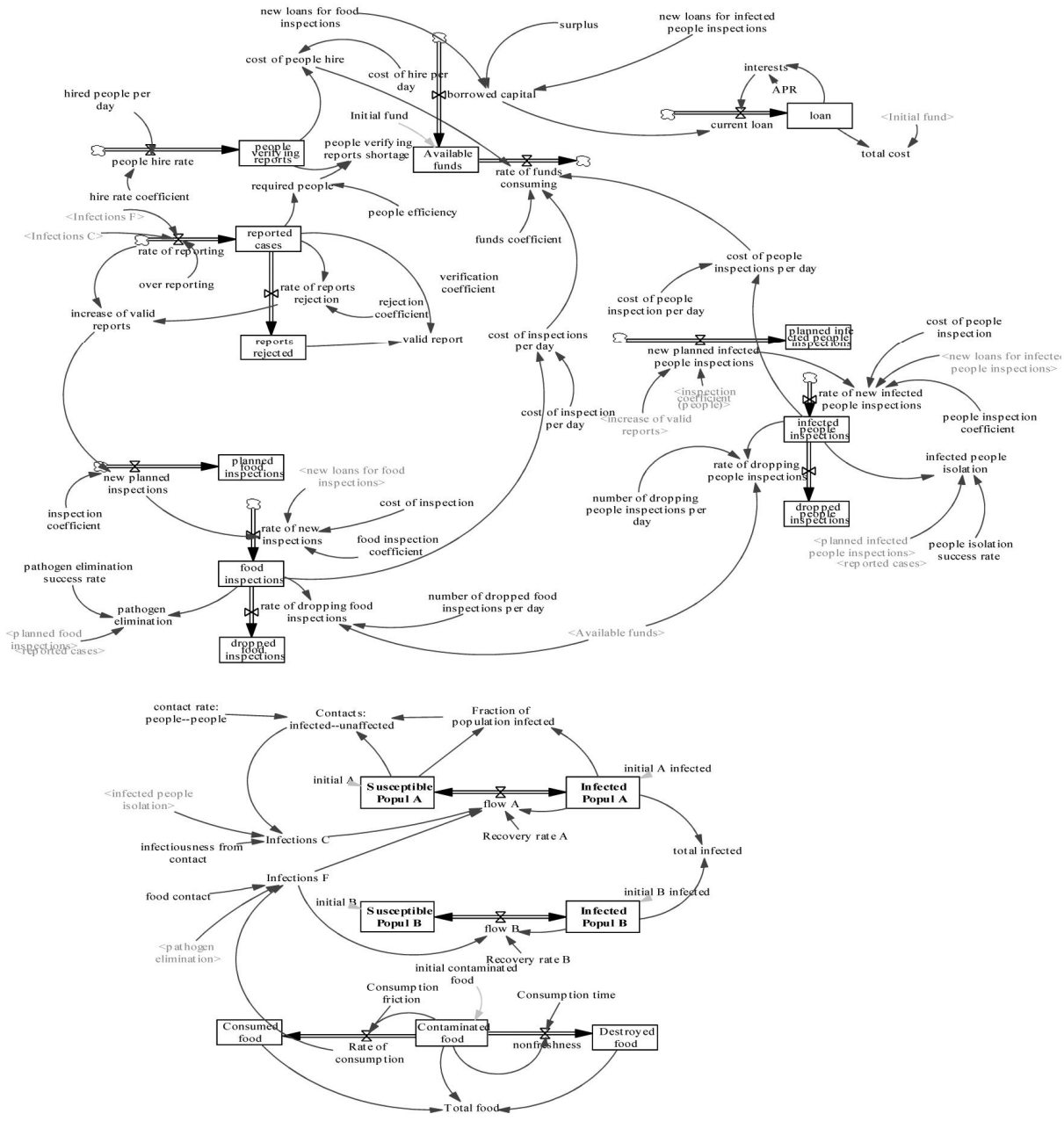
Na rysunku 1 pokazano przykładowy model zbudowany z wykorzystaniem zasad Forrestera w oparciu o aplikację Vensim.

Model zaprezentowany na rysunku 1 ma kilka elementów składowych. Pierwszy element opisuje sposób generowania raportów dotyczących rozprzestrzeniania się epidemii. Każdy raport musi być zweryfikowany w celu sprawdzenia czy potwierdzono przypadek w ramach tego samego znanego już ogniska zakażenia, czy też pojawiło się nowe ognisko.

Część modelu stanowiąca rozszerzenie klasycznego modelu prognozowania rozwoju epidemii jest pokazana na rysunku 2 (Anderson, May, 1991; Bailey, 1975; Kermack, McKendrick, 1927; Kermack, McKendrick, 1932; Kermack, McKendrick, 1933; Murray, 1993; Murray, 2001). Jest to model rozprzestrzeniania się zakażenia poprzez drogi pokarmowe. Prostokąty na rysunku modelują zmienne różniczkowe, a pozostałe elementy to parametry (lub zmienne decyzyjne w przypadku optymalizacji). W pokazanym fragmencie modelu mamy dwie zmienne Inf_Popul_A oraz Inf_Popul_B, które modelują liczbę zakażonych osób. W modelu przyjęto założenie o tym, iż infekcja może nastąpić na skutek kontaktu z zakażoną żywnością oraz z zakażoną osobą — co jest zwykle prawdą w przypadku dużej części chorób zakaźnych.

Zintegrowany model decyzyjny prezentowany w artykule zawiera w sobie także elementy modelowania decyzji przez inspektorów sanitarnych w zakresie zwalczania epidemii. Inspektorzy sanitarni zwalczają epidemię poprzez aktywne monitorowanie sytuacji w terenie, w tym wywiady z pacjentami podejrzanymi o zakażenie czynnikiem wywołującym epidemię i późniejsze działanie związane z leczeniem. Dodatkowo monitorowane są miejsca, w których przetrzymuje się podejrzaną o zakażenie żywność. Proces decyzyjny zawiera wiele reguł decyzyjnych, które powiązane są z analizą dostępnego budżetu na działania operacyjne. Reguły decyzyjne zaszyte są w zmien-

Rysunek 1
Zintegrowany model działania służb sanitarnych i rozszerzony model epidemii SIR



Źródło: opracowanie własne.

nych *rate of new inspections*, *rate of dropping food inspections*, *rate of new infected people inspections*, *rate of dropping people inspections*. Reguły decyzyjne związane z tymi zmiennymi mają podobną budowę, stąd przeanalizowana zostanie dalej jedna z takich reguł dla zmiennej *rate of new inspections.*

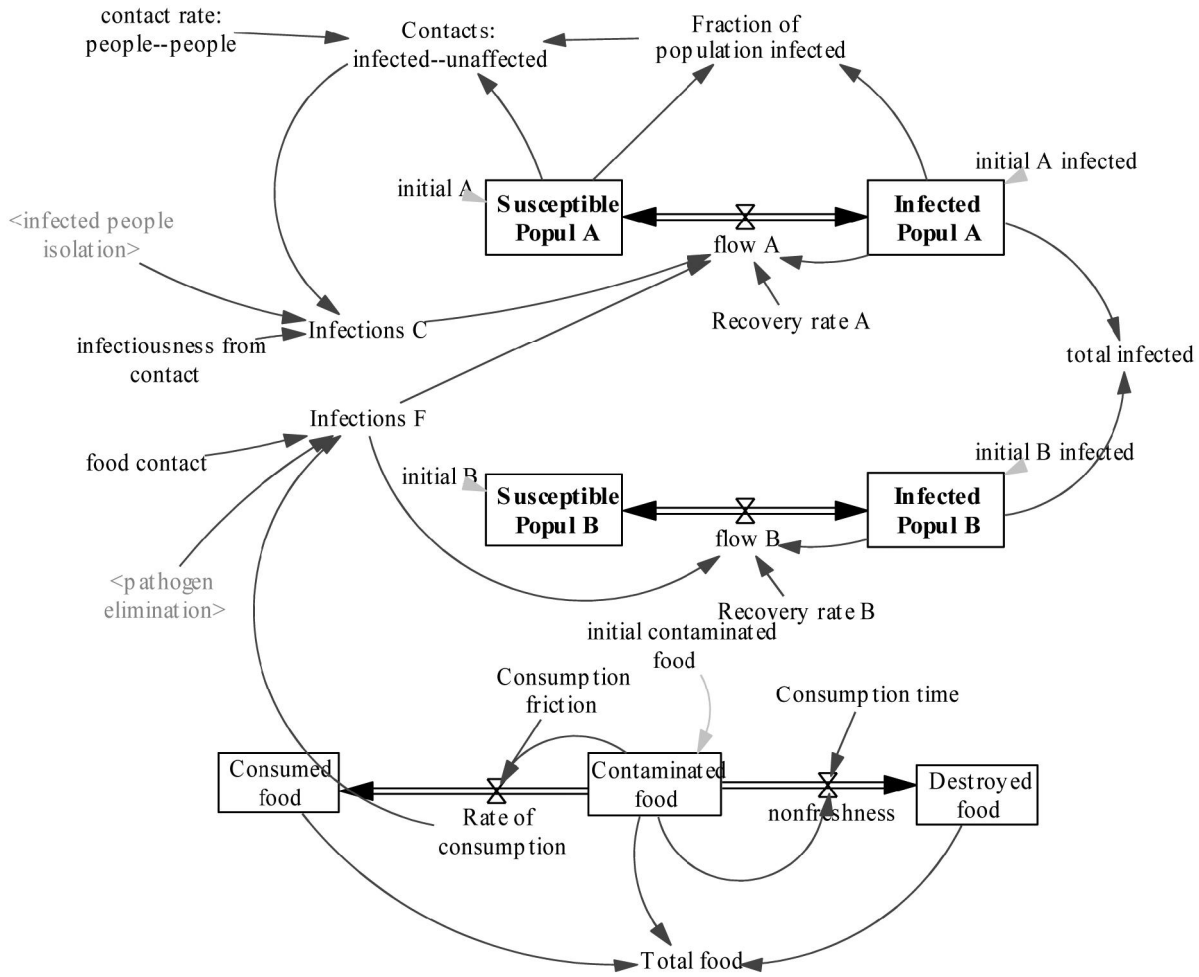
W zapisie formalnym reguła decyzyjna wygląda następująco:

rate of new infected people inspections =
people inspection coefficient* (IF THEN ELSE (INTEGER (MAX (new loans for infected people inspections/cost of people
inspection, 0)) > new planned infected people inspections,
new planned infected people inspections,
INTEGER (MAX (new loans for infected people inspections/cost of people
inspection, 0))))

Na tej podstawie określa się (biorąc pod uwagę dostępne fundusze, koszty inspekcji czy liczbę aktual-

Rysunek 2

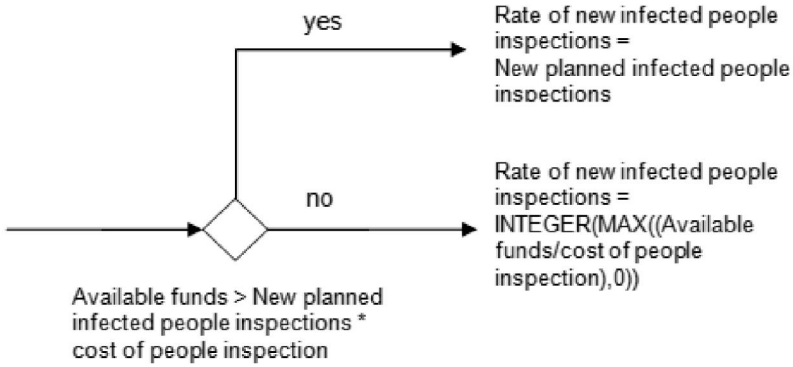
Rozszerzenie klasycznego modelu epidemii SIR



Źródło: opracowanie własne.

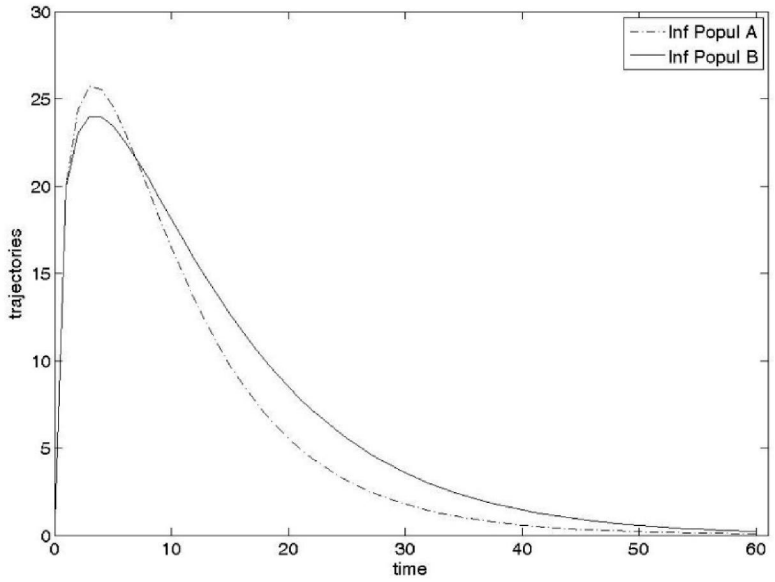
Rysunek 3

Graficzna prezentacja reguły decyzyjnej



Źródło: opracowanie własne.

Rysunek 4
Wynik symulacji modelu SIR



Źródło: opracowanie własne.

nie realizowanych inspekcji) planowaną liczbę nowych inspekcji terenowych.

Omawiany model składa się z 67 zmiennych, w tym 21 zmiennych różniczkowych. Symulacja tego modelu jest w praktyce całkowaniem numerycznym modelu złożonego z równań różniczkowo-algebraicznych przetransformowanych na równania różniczkowe zwyczajne (w tym momencie można już stosować metody Runge-Kutta do całkowania numerycznego). Wyniki przykładowej symulacji pokazano na rysunku 4.

Komponent do optymalizacji modeli

Komponent do optymalizacji spełnia dwa zadania. Po pierwsze, kalibruje model. Po drugie, używa się go do wyznaczania decyzji optymalnych w zakresie działań inspektorów sanitarnych.

Optymalizacja działań służb sanitarnych

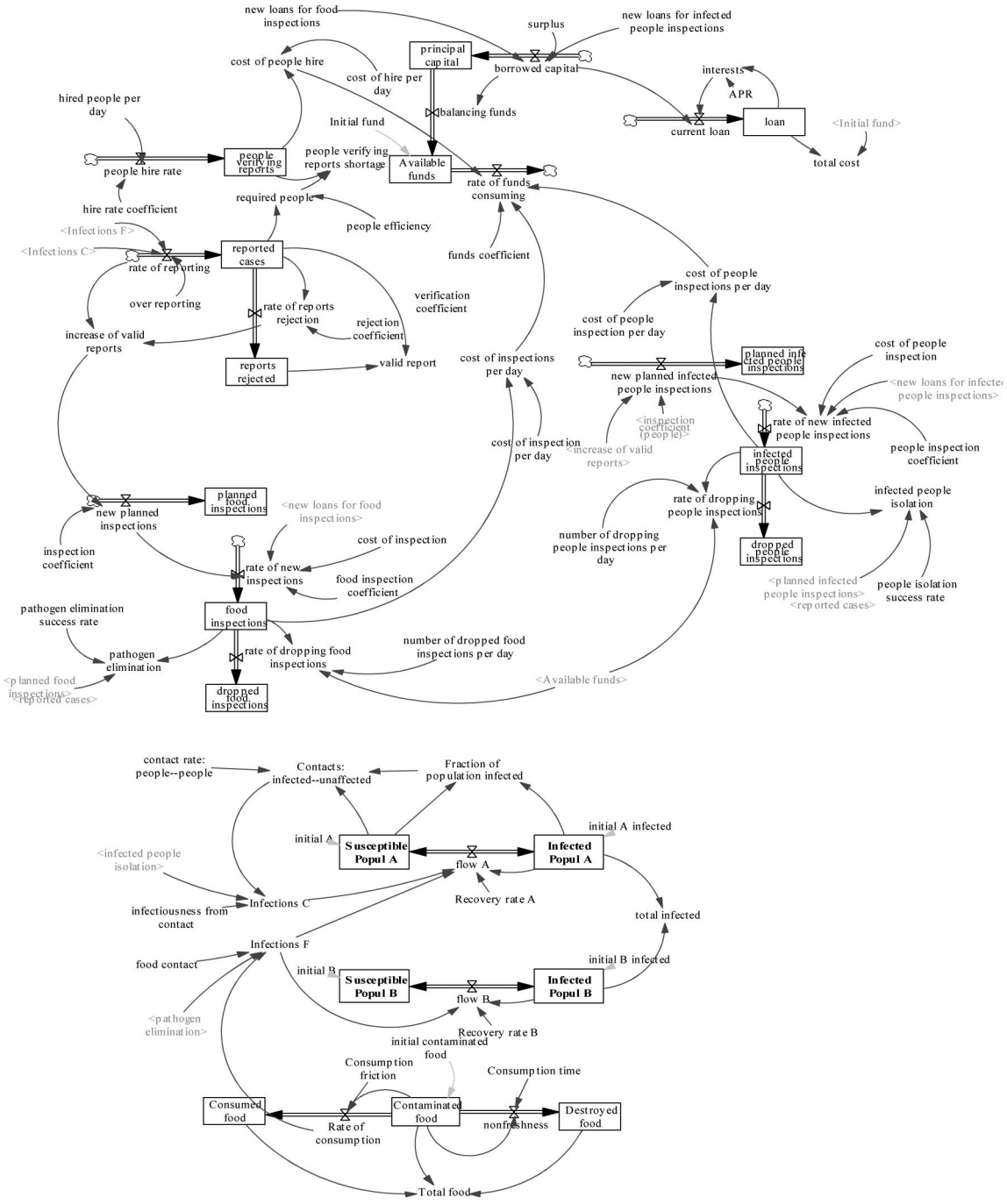
Nasz system może być użyty do optymalizacji zadań dla służb sanitarnych. Realizuje się to z wykorzystaniem modeli zbudowanych do symulacji działań służb na bazie metodyki Forrestera. W pierwszym kroku eliminuje się równania algebraiczne, aby pozostały same równania różniczkowe zwyczajne. Wykonuje to podsystem DOML File Generator. Format DOML jest rozpoznawany przez serwer IDOS, który rozwiązuje zadanie na bazie wygenerowanego skryptu z modelem problemu. DOML jest rozwinięciem języka Modelica, który stanowi standard języków symulacji procesów. DOML File Generator daje jedynie część opisu zadania optymalizacji w postaci równań różniczkowych. Pozostałe elementy — zmienne sterujące, funkcję celu, ograniczenia — można zdefiniować w narzędziu Optimizer GUI.

Skrypt w języku DOML jest przekazywany do serwera IDOS w celu rozwiązania zadania optymalizacji (Pytlak, Tarnawski, Fajdek, Stachura, 2013). IDOS przypomina narzędzie do rozwiązywania zadań statycznych NEOS i analogicznie jak ono wykorzystuje kody optymalizacyjne otwarte. IDOS jest wyposażony w bardzo zaawansowane algorytmy optymalizacji oparte na dyskretyzacji a priori dla równań różniczkowych zwyczajnych, solwery dla równań sprzężonych, metody oparte o procedury zwane *shooting procedures*. Każda z tych metod może zostać użyta do rozwiązania problemu działania służb sanitarnych (zobacz kod z opisem metody po słowie *annotation*). IDOS wyposażony jest dodatkowo w metody automatycznego różniczkowania, które ułatwiają rozwiązywanie zadania optymalizacji.

Przykład optymalizacji dynamicznej

W celu zilustrowania dziania serwera IDOS użyty zostanie model zdefiniowany w poprzedniej części artykułu. Eliminujemy reguły decyzyjne z modelu symulacyjnego. W ich miejsce pojawiają się zmienne decyzyjne. Definiujemy je jako zmien-

Rysunek 5
Model ze zmiennymi sterującymi



Źródło: opracowanie własne.

ne wejściowe zadania, co jest uwidocznione w skrypcie DOML. W celu minimalizacji liczby zakażonych osób populacji A i B dodajemy nową zmienną (dwa ostatnie równania z sekcji *equation* z rysunku 6 i 7):

der (Total_Inf_Pop_A) — Inf_Pop_A = 0
der (Total_Inf_Pop_B) — Inf_Pop_B = 0.

Model optymalizacyjny ma wiele ograniczeń w postaci nierówności, które muszą być spełnione w każdym kroku optymalizacji. Wśród tych ograniczeń są ograniczenia zabezpieczające przed przekroczeniem przyjętych z góry wartości. Model optymalizacyjny posiada 21 równań różniczkowych, a horyzont optymalizacji został ustawiony na maksymalnie 60 dni.

Zadanie rozwiązane zostało przez pakiet optymalizacyjny Ipopt (Wachter, 2002), który jest dostępny poprzez solwer *olado* w IDOS. Wyniki pokazano na rysunku 8. Jak można było przypuszczać, wynik opty-

Rysunek 6
Zadanie optymalizacji w DOML — definicje

```
package Epidemic_olado

optimization Epidemic_olado_opt(
objective = Total_Inf_Pop_A(finalTime) + Total_Inf_Pop_B(finalTime),
startTime = 0.0,
finalTime = 60.0)

parameter Real cost_of_food_inspections = 100;
parameter Real surplus = 500;
parameter Real cost_of_infected_people_inspection = 100;
parameter Real funds_coefficient = 1;
parameter Real cost_of_hire_per_day = 0.03;
. .          .          .          .          .          .
parameter Real recovery_rate_A = 0.2;
parameter Real recovery_rate_B = 0.1;
parameter Real Consumption_time = 15;
Real principal_capital(start = 0.0);
Real available_funds(start = 50.0);
Real people_veryfing_reports(start = 1.0);
Real loan(start = 0.0);  Real reported_cases(start = 0.0);  Real reports_rejected(start = 0.0);
Real planned_infected_people_inspections(start = 0.0);
Real infected_people_inspections(start = 0.0);
Real dropped_people_inspections(start = 0.0);
Real planned_food_inspections(start = 0.0);
Real food_inspections(start = 0.0);
Real dropped_food_inspections(start = 0.0);
Real Susc_Pop_A(start = 1000.0);
Real Inf_Pop_A(start = 0.0);  Real Susc_Pop_B(start = 200.0);  Real Inf_Pop_B(start = 0.0);
Real Contaminated_food(start = 800.0);
Real Destroyed_food(start = 1.0);  Real Consumed_food(start = 0.0);
Real Total_Inf_Pop_A(start = 0.0); Real Total_Inf_Pop_B(start = 0.0);
input Real hired_people_per_day (initialGuess = 5, min=0, max=30);
input Real new_food_inspections_per_day (initialGuess = 5, min=0, max=30);
input Real new_infected_people_inspections_per_day (initialGuess= 5, min=0, max=30);
input Real number_of_dropped_food_inspections_per_day (initialGuess= 5, min=0, max=30);
input Real number_of_dropping_people_inspections_per_day (initialGuess= 5, min=0, max=30);

annotation(solver="olado", steps="60", solver_option="method=17",
solver_option="inifile=olado.ini",solver_option="eps=1e-6", solver_option="max_iter=1000");

equation
der(principal_capital) - (((cost_of_food_inspections) * (new_food_inspections_per_day)) + (surplus)) +(
```

Źródło: opracowanie własne.

malizacji w zakresie liczby zakażonych osób jest lepszy niż wynik symulacji procesu. Fakt, że liczba zakażonych osób nie zmalała do zera, wiąże się z podatnością populacji na ponowne zakażenia.

Optymalizacja oparta na metodach CLP

Programowanie z ograniczeniami (ang. *Constraint Logic Programming* — CLP) stanowi alternatywny i bardzo wydajny mechanizm wykorzystywany do modelowania i rozwiązywania zadań harmonogramowania. Podstawową zaletą CLP jest deklaratywność, czyli możliwość formułowania zadania optymalizacji bez konieczności implementacji metod rozwiązania tego zadania. CLP bazuje na modelowaniu zadania jako problemu spełnienia ograniczeń CSP (ang. *Constraint Satisfaction Problem*). Ograniczenia zależą ściśle od dziedzin zmiennych, których dotyczą. Najczęściej spotykane dziedziny to dziedzina liczb całkowitych i rzeczywistych.

Główną metodą poszukiwania rozwiązania zadania jest mechanizm propagacji ograniczeń. Narzędzia implementujące metody CLP oparte są głównie na modyfikacji Prologu (CHIP, Eclipse — CISCO Systems, 2006; SICStus Prolog).

W niniejszej pracy zadania CLP rozwiązywano z wykorzystaniem mechanizmów ECLiPSe CLP. Metody wnioskowania w logice predykatów pierwszego rzędu, na jakiej oparta jest filozofia CLP, podane zostały przez T. Fruwirtha i S. Abdennadhera (Fruwirth, Abdennadher, 2003) Na rysunkach 9 i 10 zaprezentowano najważniejsze elementy definiujące problem optymalizacyjny w modelu CLP.

Rysunek 7

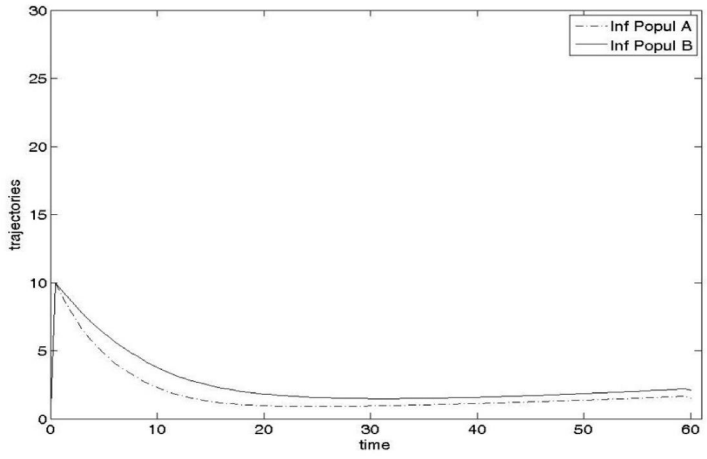
Zadanie optymalizacji w DOML

```
der (loan) — ( ( ( ( (cost_of_food_inspections) * (new_food_inspections_per_day)) + (surplus)) + ( (cost_of_infected_people_inspection) * (new_infected_people_inspections_per_day))) + ( (APR) * (loan))) - ( ( (cost_of_food_inspections) * (number_of_dropped_food_inspections_per_day)) + ( (cost_of_infected_people_inspection) * (number_of_dropping_people_inspections_per_day))) =0;

der (reported_cases) — ( (over_reporting) * ( ( ( ( ( (contact_rate_people_people) * (Susc_Pop_A)) * ( (Inf_Pop_A) / ( (Inf_Pop_A) + (Susc_Pop_A)))) * (infectiousness_from_contact)) * ( (1) - ( ( (people_isolation_success_rate) * (infected_people_inspections)) / ( (planned_infected_people_inspections) + (1))))) + ( ( ( (Consumption_friction) * (Contaminated_food)) * (food_contact)) * ( (1) - ( ( (pathogen_elimination_succes_rate) * (food_inspections)) / ( (planned_food_inspections) + (1)))))))) - ( (rejections_coefficient) * (reported_cases)) =0;

der (reports_rejected) — ( (rejections_coefficient) * (reported_cases)) - (0) =0;
der (planned_infected_people_inspections) — ( ( ( (over_reporting) * ( ( ( ( ( (contact_rate_people_people) * (Susc_Pop_A)) * ( (Inf_Pop_A) / ( (Inf_Pop_A) + (Susc_Pop_A)))) * (infectiousness_from_contact)) * ( (1) - ( ( (people_isolation_success_rate) * (infected_people_inspections)) / ( (planned_infected_people_inspections) + (1))))) + ( ( ( (Consumption_friction) * (Contaminated_food)) * (food_contact)) * ( (1) - ( ( (pathogen_elimination_succes_rate) * (food_inspections)) / ( (planned_food_inspections) + (1)))))))) - ( (rejections_coefficient) * (reported_cases))) * (inspection_coefficient_people)) - (0) =0;

der (infected_people_inspections) — ( (people_inspection_coefficient) * (new_infected_people_inspections_per_day)) - (number_of_dropping_people_inspections_per_day) =0;
der (dropped_people_inspections) — (number_of_dropping_people_inspections_per_day) - (0) =0;

der (planned_food_inspections) — ( (inspections_coefficient) * ( ( (over_reporting) * ( ( ( ( ( (contact_rate_people_people) * (Susc_Pop_A)) * ( (Inf_Pop_A) / ( (Inf_Pop_A) + (Susc_Pop_A)))) * (infectiousness_from_contact)) * ( (1) - ( ( (people_isolation_success_rate) * (infected_people_inspections)) / ( (planned_infected_people_inspections) + (1))))) + ( ( ( (Consumption_friction) * (Contaminated_food)) * (food_contact)) * ( (1) - ( ( (pathogen_elimination_succes_rate) * (food_inspections)) / ( (planned_food_inspections) + (1)))))))) - ( (rejections_coefficient) * (reported_cases)))) - (0) =0;

der (Inf_Pop_A) — ( ( ( ( ( ( (contact_rate_people_people) * (Susc_Pop_A)) * ( (Inf_Pop_A) / ( (Inf_Pop_A) + (Susc_Pop_A)))) * (infectiousness_from_contact)) * ( (1) — ( ( (people_isolation_success_rate) * (infected_people_inspections)) / ( (planned_infected_people_inspections) + (1))))) + ( ( ( (Consumption_friction) *
(Contaminated_food)) * (food_contact)) * ((1) - (((pathogen_elimination_succes_rate) * (food_inspections)) / ((planned_food_inspections) + (1)))))) -
((recovery_rate_A)*(Inf_Pop_A)))-(0)=0;
. .  .  .  .  .  .  .  .
der(Inf_Pop_B) — (((((Consumption_friction) * (Contaminated_food)) * (food_contact))*((1) - (((pathogen_elimination_succes_rate) * (food_inspections)) /
((planned_food_inspections)+(1))))) - ((recovery_rate_B)*(Inf_Pop_B)))-(0)=0;

der(Contaminated_food) — (((Contaminated_food)/(Consumption_time)) + ((Consumption_friction)*(Contaminated_food)))=0;
der(Destroyed_food) — ((Contaminated_food)/(Consumption_time))-(0)=0;
der(Consumed_food) — ((Consumption_friction)*(Contaminated_food))-(0)=0;
der(Total_Inf_Pop_A) — Inf_Pop_A =0;  der(Total_Inf_Pop_B) - Inf_Pop_B =0;

constraint
```

Źródło: opracowanie własne.

Rysunek 8

Przykładowy wynik optymalizacji



Źródło: opracowanie własne.

Rysunek 9
Definicja problemu w CLP

```
optimize_SIR(SimTime) :-

  /* SIR state variables */
  dim(ConsumedFood,[NbPeriods]),
  dim(ContaminatedFood,[NbPeriods]),
  dim(DestroyedFood,[NbPeriods]),
  …
  /* initial values */
  ContaminatedFood[1] $= 9000.0,
  ConsumedFood[1]     $= 0.0,
  …
  /* constraints FOOD */
  ( for(T,1,NbPeriods), param(RateOfConsumption,ConsumptionFriction,ContaminatedFood) do
     RateOfConsumption[T] $= eval(ConsumptionFriction*ContaminatedFood[T])   ),
  ( for(T,1,NbPeriods), param(Nonfreshness,ConsumptionTime,ContaminatedFood) do
     Nonfreshness[T] $= eval(ContaminatedFood[T]/ConsumptionTime)   ),
  ( for(T,1,NbPeriods), param(TotalFood,ContaminatedFood,DestroyedFood,ConsumedFood) do
     TotalFood[T] $= ContaminatedFood[T] + DestroyedFood[T] + ConsumedFood[T]     ),
  ( for(T,1,NbPeriods-1), param(ConsumedFood,RateOfConsumption) do
     eval(ConsumedFood[T+1]) - eval(ConsumedFood[T]) $= eval(RateOfConsumption[T])   ),
  ( for(T,1,NbPeriods-1), param(DestroyedFood,Nonfreshness) do
     eval(DestroyedFood[T+1]) - eval(DestroyedFood[T]) $= eval(Nonfreshness[T])   ),
  ( for(T,1,NbPeriods-1), param(ContaminatedFood,Nonfreshness,RateOfConsumption) do
    eval(ContaminatedFood[T+1]) - eval(ContaminatedFood[T]) $=  - eval(Nonfreshness[T]) - eval(RateOfConsumption[T])   ),

  /* constraints Infected B Group */
  ( for(T,1,NbPeriods), param(InfectionsF,RateOfConsumption,FoodContact,PathogenElimination) do
     InfectionsF[T] $= RateOfConsumption[T]*FoodContact*(1-PathogenElimination[T])   ),
  ( for(T,1,NbPeriods), param(FlowB,InfectionsF,RecoveryRateB,InfectedPopulB) do
     FlowB[T] $= InfectionsF[T] - RecoveryRateB * InfectedPopulB[T]   ),
  ( for(T,1,NbPeriods-1), param(SusceptiblePopulB,FlowB) do
     SusceptiblePopulB[T+1] - SusceptiblePopulB[T] $= -1 * eval(FlowB[T])   ),
  ( for(T,1,NbPeriods-1), param(InfectedPopulB,FlowB) do
     %InfectedPopulB[T+1] - InfectedPopulB[T] $=< eval(FlowB[T]),
     InfectedPopulB[T+1] - InfectedPopulB[T] $= eval(FlowB[T])   ),

  /* constraints Infected A Group */
  ( for(T,1,NbPeriods), param(FractionOfPopulationInfected,InfectedPopulA,SusceptiblePopulA) do
     FractionOfPopulationInfected[T] $= InfectedPopulA[T]/(InfectedPopulA[T]+SusceptiblePopulA[T])     ),
  ( for(T,1,NbPeriods),
  param(ContactsInfectedUnaffected,ContactRatePeoplePeople,SusceptiblePopulA,FractionOfPopulationInfected) do
     ContactsInfectedUnaffected[T] $= ContactRatePeoplePeople*SusceptiblePopulA[T]*FractionOfPopulationInfected[T]   ),
```

Źródło: opracowanie własne.

Zaprezentowany przykład minimalizuje liczbę zakażonych w populacji A i B. Wykorzystuje się metodę optymalizacji opartą na drzewach Branch&Bound.

Rysunek 10
Definicja problemu w CLP

```
( for(T,1,NbPeriods),
   fromto(0,In1,In1+InfectedPopulA[T],InfectedA),param(InfectedPopulA) do
    true  ),

( for(T,1,NbPeriods),
   fromto(0,In1,In1+InfectedPopulB[T],InfectedB),param(InfectedPopulB) do
    true  ),

TotalCosts $= eval(InfectedB) + eval(InfectedA),

%
% search
%
term_variables([
  ContaminatedFood,ConsumedFood,DestroyedFood,
  InfectedPopulB,SusceptiblePopulB,
  PathogenElimination,
  InfectedPopulA,SusceptiblePopulA,
  InfectedPeopleIsolation,
  TotalCosts], Vars),

 bb_min(search(Vars,0,smallest,indomain,complete,[backtrack(Backtracks)]),TotalCosts,
          bb_options{strategy:continue}),
```

Źródło: opracowanie własne.

Podsumowanie

W artykule opisano system wspomagania działań inspektorów sanitarnych. Zaprezentowano także kilka nowych elementów, niespotykanych w literaturze światowej. Poszerzono opis modelu o działania inspektorów sanitarnych w zakresie ograniczania skutków epidemii. Modelowanie oparto o metodykę Forrestera. Model wykorzystano w praktyce do rozwiązania zadania optymalizacji działań służb sanitarnych. Serwer optymalizacyjny IDOS jest dostępny w sieci Internet ze stron Politechniki Warszawskiej.

Literatura

Anderson, R.M., May. (1991). (ed.). *Infectious Diseases of Humans: Dynamics and Control.* Oxford: Oxford University Press.

Bailey, N.T.J. (1975). The Mathematical Theory of Infectious Diseases. London: Griffin.

Fruwirth T., Abdennadher S. (2003). *Essentials of constraint programming*. Springer.

JModelica. org. User Guide. Version 1.8. (2012). Modelon AB.

Kermack, W.O., McKendrick, A.G. (1927). Contributions to the mathematical theory of epidemics. *Proceedings of the Royal Society of London*. Series A, 115: 700–721.

Kermack, W.O., McKendrick, A.G. (1932). Contributions to the mathematical theory of epidemics. *Proceedings of the Royal Society of London*. Series A, 138: 55–83.

Kermack, W.O., McKendrick, A.G. (1933). Contributions to the mathematical theory of epidemics. *Proceedings of the Royal Society of London*. Series A, 141: 94–122.

Murray, J.D. (1933). *Mathematical biology* (I). *An introduction*. Berlin, Heidelberg, New York: Springer-Verlag.

Murray, J.D. (2001). *Mathematical biology* (II). *Spatial models*. Berlin, Heidelberg, New York: Springer-Verlag.

Pytlak, R. (1999). Numerical Methods for Optimal Control Problems with State Constraints. Lecture Notes in Mathematics, 1707. Springer-Verlag.

Pytlak, R., Tarnawski, T., Fajdek, B., Stachura, M. (2013). Interactive Dynamic Optimization Server (IDOS) — Connecting one Modeling Language with Many Solvers, Optimization Methods & Solvers, in print.

Sterman, J.D. (2000). *Business Dynamics.* McGraw-Hill.

Wachter, A. (2002). *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*. Pittsburgh, PA, USA: Carnegie Mellon University.

Annals of Mathematics and Artificial Intelligence. 2002, 34: 259–269.

Wallace, M. (2005). Hybrid algorithms, local search and ECLiPSe. CP Summer School.

Wallace, M., Schimpf, J. (2002). Finding the right hybrid algorithm — A combinatorial meta-problem. *Annals of Mathematics and Artificial Intelligence*, (34), 259–269.

Zawadzki, T., Pytlak, R. (2011). Extending System Dynamics Approach to Higher Index DAE's. Proceedings of the System Dynamics Society 2011 Conference July 24–26. Washington, USA.