Spis treści • *Contents*

RECENZOWANE ARTYKUŁY NAUKOWE
REVIEWED SCIENTIFIC ARTICLES

Realizacja szyfru „bezkluczowego” `c80k395` do kryptograficznej ochrony załączników poczty elektronicznej w środowisku Maple

Maple implementation of „Keyless” Cipher **`c80k395`** *for cryptographic protection of e-mail enclosures*

Czesław Kościelny[1]

Treść. Opisano przykład aplikacji typu workhsheet, uruchamianej w środowisku Maple i realizującej procedury „bezkluczowego” szyfrowania i deszyfrowana plików za pomocą symetrycznego szyfru plikowego **`c80k395`**. Aplikacja posiada prosty graficzny interfejs użytkownika i przeznaczona jest głównie do kryptograficznej ochrony załączników poczty elektronicznej.

Słowa kluczowe: kodowanie transportowe, funkcja Maple **`convert/base`**, szyfrowanie symetryczne plików.

Abstract. An example of a Maple worksheet application which performs the „keyless” file encryption or decryption by means of the symmetric cipher **`c80k395`** has been presented. The application has a simple graphical user interface and may be used mainly for cryptographic protection of e-mail enclosures.

Keywords: Base 64 Encoding Scheme, Maple **`convert/base`** built-in function, Symmetric File Encryption.

1. Wstęp

Autor zauważył, że nieliniowe przekształcenia stosowane w systemach kodowania transportowego, opisanego w dokumencie RFC 4648, można zastosować do konstrukcji nowej rodziny szyfrów, które nie są ani szyframi strumieniowymi, ani szyframi blokowymi, aktualnie stosowanymi w kryptografii klasycznej [1]. Aby taki nowy szyfr skonstruować wystarczy algorytmy szyfrowania i deszyfrow ania zastąpić odpowiednio zmodyfikowanymi algorytmami kodowania i dekodowania transportowego. Jak wiadomo, procedura realizująca algorytm kodowania transportowego według tymczasowej normy internetowej RFC 4648 [2] generuje plik zawierający ściśle określony zestaw znaków siedmiobitowego kodu ASCII a każdemu znakowi odpowiada ustalona sekwencja 4, 5 lub 6 bitów. Procedura dekodowania poprawnie dekoduje taki plik zakodowany, w którym zachowane są reguły kodowania transportowego. Ten określony i skonfigurowany zgodnie z normą RFC zestaw znaków nazywa się alfabetem. Wyposażając procedury kodowania i dekodowania transportowego w parametr formalny, zwany kluczem, umożliwiający zmianę stosowanego alfabetu, otrzymuje się wygodne narzędzie do szyfrowania i deszyfrowania plików. Tworzone w ten sposób szyfry można nazwać symetrycznymi szyframi plikowymi, ponieważ procedury realizujące algorytm szyfrujący i algorytm deszyfrujący posiadają dwa parametry formalne: nazwę pliku do zaszyfrowania lub zdeszyfrowania oraz klucz, który jak zwykle w przypadku szyfru symetrycznego jest taki sam dla procedury szyfrującej oraz dla procedury deszyfrującej. Zaproponowano też, aby szyfrom nadawać nazwę **`cxky`**, gdzie po literze **`c`**, oznaczającej znak (character), umieszczona jest liczba znaków **`x`**, występujących w pliku zaszyfrowanym, a litera **`k`** (key) poprzedza liczbę bitów **`y`**, charakteryzującą moc szyfru. W opisywanym programie szyfrująco-deszyfrującym zastosowano funkcję biblioteczną programu Maple o nazwie **`convert/base`**. Jest to narzędzie bardziej universalne niż opisywane w normie RFC 4848 kodowanie transportowe, pozwala bowiem stosować dowolne wartości bazy. Poza tym ustalono, że w zaszyfrowanym pliku będzie występować 10 cyfr oraz 52 duże i małe litery polskiego alfabetu, łącznie 80 znaków drukowalnych. W takiej sytuacji tajnym kluczem szyfrującym jest lista, występująca w kodzie źródłowym pod nazwą **`a`**, zawierająca 80 wartości bajtowych znaków pokazanych na rys. 1. Zatem przestrzeń klucza posiada 80! elementów, czyli w przybliżeniu

$$7.156945705\ 10^{118}.$$

W zapisie binarnym jest to liczba 395-bitowa, można więc powiedzieć, że aplikacja stosuje tajny klucz o tej właśnie długości. Aplikację można zaprogramować albo w sposób tradycyjny, publikując jej kod źródłowy i stosując tajny klucz, albo klucz „wmontować" do aplikacji, którą wtedy należy stosownie chronić. W tym drugim przypadku aplikację można nazwać szyfrem „bezkluczowym", ponieważ to narzędzie realizuje procedury szyfrowania i deszyfrowania oraz pełni też rolę tajnego klucza. Ten „bezkluczowy” sposób szyfrowany jest wygodny dla wielu użytkowników i na ogół stosowany jest w wojsku. Aplikacja została napisana w języku Maple, a jej kod źródłowy oszczędnie

1Wydział Informatyki, Wrocławska Wyższa Szkoła Informatyki Stosowanej, ul. Wejherowska 28, 54-239 Wrocław, ckoscielny@horyzont.eu

zapisany zajmuje około 60% jednej strony formatu A4. Dlatego czytelnicy, którzy są zainteresowani szczegółami aplikacji mogą ściągnąć dostępny w internecie [3] plik **c80k395.mw.** W artykule zostanie jeszcze opisany skrótowo tylko kod źródłowy.

2. Kod źródłowy

Zadanie szyfru „bezkluczowego" realizuje aplikacja w postaci pliku **c80k395.maplet**, zapisana w najnowszej wersji Maple (Rys. 1), ale aplikację można także uruchomić w wersjach starszych, np. Maple 14.

Rys. 1. Wersja Maple, w której zapisano program [3]
Fig. 1. The Maple version in which the program [3] is implemented

W programie zadeklarowano zmienne **a**, **self2ed**, **ef**, **df**, przy czym:
a − zmienna w postaci listy liczb typu byte, „montująca” tajny klucz w procedurach szyfrującej i deszyfrującej,
self2ed − procedura do wybierania plików, które mają być szyfrowane lub deszyfrowane,
ef − procedura szyfrująca,
df − procedura deszyfrująca.

Aby wydrukować lub obejrzeć kod źródłowy należy plik **c80k395.mw** otworzyć w sesji Maple. Program jest nieskomplikowany i wyjątkowo prosty w obsłudze: użytkownik nie musi znać języka Maple lecz powinien umieć posługiwać się myszą i jej klawiszami. Jak wspomniano, aplikację zaprogramowano w postaci pliku **c80k395.mw.** Wystarczy ten plik dwukrotnie kliknąć w komputerze z zainstalowanym programem Maple i wtedy zostanie wygenerowane okno interfejsu graficznego, pokazane na rys. 2. Po kliknięciu trójkąta obok napisu `Pomoc` ukaże się krótki opis aplikacji i sposobu jej użycia. Aplikacja musi posiadać uprawnienia do modyfikowania, usuwania, odczytu i zapisu przetwarzanych plików.

Rys. 2. Interfejs graficzny użytkownika aplikacji
Fig. 2. The graphical user interface of the application

Po uruchomieniu aplikacji użytkownik wybiera plik do zaszyfrowania lub zdeszyfrowania, a po zakończeniu pracy programu pojawi się szczegółowy opis wykonanych operacji.
Jeśli Czytelnik nie posiada zainstalowanego programu Maple i ma zamiar wypróbować jak działa opisana aplikacja, powinien zainstalować okrojoną, bezpłatną wersję programu Maple, czyli program Maple Player (http://www.maplesoft.com/products/maple/mapleplayer/). W tym środowisku działa aplikacja **80k395mp.mw**, nieco mniej wygodna dla użytkownika.

3. Podsumowanie i wnioski

Przedstawiono zaawansowaną aplikację zrealizowaną w środowisku Maple, która powinna być stosowana przede wszystkim do kryptograficznej ochrony załączników poczty elektronicznej (w postaci plików o dowolnym formacie) przed nieupoważnionym dostępem, ponieważ wygenerowane za pomocą aplikacji kryptogramy plików zawierają wyłącznie 80 znaków drukowalnych 7-bitowego kodu ASCII a takie załączniki są akceptowane przez wszystkie systemy internetowej poczty. Aplikacja używa szyfru z 395-bitowym kluczem, zaś algorytmy kryptograficzne oparto na transformacjach używanych w funkcji bibliotecznej **convert/base** programu Maple. Dzięki temu szyfr jest w wysokim stopniu randomizowany i jedynym znanym aktualnie autorowi sposobem łamania szyfru jest wypróbowywanie po kolei wszystkich kluczy kryptograficznych. Aplikacja jest wersją demonstracyjną, dużo wolniejszą od rozwiązań zrealizowanych w językach kompilowanych , mimo to nie stwarza istotnych problemów, jeśli rozmiar szyfrowanych plików nie przekracza 10 kB. Warto zauważyć, że rozmiar wygenerowanych plików zaszyfrowanych za pomocą aplikacji jest około 26,5% większy od rozmiaru pliku niezaszyfrowanego, podczas gdy aplikacja [1] generuje pliki kryptogramów z kluczem o 99 bitów krótszym i o 33% większe od plików niezaszy-

frowanych. Jest rzeczą oczywistą, że podobne aplikacje można realizować dla dowolnych wartości bazy, akceptowanych przez funkcję biblioteczną **`convert/base`** za pomocą których można także szyfrować pliki, zapamiętywane w dyskowych bazach danych. W tym przypadku warto zapisane w języku Maple algorytmy zrealizować w języku kompilowanym.

Podziękowanie

Autor pragnie podziękować anonimowemu recenzentowi za cenne uwagi, które pomogły usunąć usterki programu.

Literatura (References)

[1] C. Kościelny, Base 64 "Keyless" File Encryption, http://www.maplesoft.com/applications/view.aspx?SID=145918, 04.2013

[2] S. Josefsson, The Base16, Base32, and Base64 Data Encodings, http://www.h-online.com/nettools/rfc/rfcs/rfc4648.shtml , 10.2006

[3] C. Kościelny, Aplikacje **`c80k395.mw`** i **`c80k395mp.mw`**, link na portalu WWSIS: www.wydawnictwo.horyzont.eu/podstrony/publikacje.html

Operacyjne i analityczne bazy danych wspomagające procesy decyzyjne w zarządzaniu uczelnią wyższą

Transactional and analytical databases to support decision-making in the management of a university

Swietłana Lebiediewa[1]

Treść. Przedstawiono charakterystykę użytkowników bazy danych i potrzeby różnych grup użytkowników. Omówiono zawartość informacyjną bazy danych. Omówiono rolę bazy danych w zarządzaniu szkołą wyższa dotyczącą dwóch aspektów: zarządzania operacyjnego i podejmowania decyzji strategicznych. Przytoczono przykłady informacji potrzebnej do podejmowania decyzji operacyjnych i strategicznych.

Słowa kluczowe: baza danych, zarządzanie, wyższa szkoła niepaństwowa

Abstract. The characteristics of database users and the needs of different user groups are presented. The information content of the database is discussed. Database role in the management of university on two aspects: operational and strategic decision-making is discussed. Examples of information needed to make operational and strategic decisions are quoted.

Keywords: database, management, non-public university

1. Wstęp

Warunkiem sprawnego funkcjonowania organizacji jest posiadanie informacji. Zadaniem systemów informatycznych jest wspomaganie zarządzania na każdym szczeblu, ogólnie przyjęta jest następująca klasyfikacja systemów informatycznych wspomagających zarządzanie [1,2]:

- Systemy transakcyjne,
- Systemy informacyjne,
- Systemy wspomagające podejmowanie decyzji,
- Systemy informowania kierownictwa.

Przedmiotem pracy jest przedstawienie baz danych wspomagających procesy decyzyjne zarówno na szczeblu taktyczno-operacyjnym (Transaction Processing lub Operational System systemy OLTP) jak i systemów wspomagających podejmowanie decyzji strategicznych (strategic decisions suport systems lub Online Analytical Processing systemy OLAP).

2. Użytkownicy i zawartość informacyjna bazy danych

Użytkownikami bazy danych są: kierownictwo szkoły (kanclerz, wicekanclerze, rektor, prorektorzy, dziekani, prodziekani), pracownicy administracyjni, pracownicy dziekanatów i działu finansowego, wykładowcy, studenci, pracownicy działu promocji i marketingu. Każda z grup użytkowników potrzebuje na ogół odmiennych informacji. Kierownictwo szkoły potrzebuje informacji o liczbie zgłoszeń na poszczególne kierunki, wykonaniu pensum i aktualnym obciążeniu dydaktycznym wykładowców, liczbie grup laboratoryjnych na poszczególnych latach i poszczególnych kierunkach, wynikach sesji, liczbie absolwentów w poszczególnych latach. Pracownicy dziekanatu mogą chcieć uzyskać informacje o danych personalnych studenta, wyniku sesji, a także informacje, czy student X może być wpisany na dany semestr, czy student ma zaległości, kto jest promotorem studenta, ile dyplomantów ma wykładowca. Wykładowcom jest potrzebna informacja o obciążeniu dydaktycznym w danym semestrze lub w roku szkolnym, rozkład zajęć, listy studentów. Studenci potrzebują informacji o planie studiów na poszczególnych specjalnościach, o rozkładzie zajęć, o egzaminach i zaliczeniach w danym semestrze, o wynikach egzaminów i zaliczeń, o wykładowcach poszczególnych przedmiotów. Dział organizacji studiów musi mieć informację o salach wykładowych i laboratoriach, liczbie grup studenckich, liczbie osób w grupach, przedmiotach na poszczególnych semestrach, preferencji wykładowców prowadzących poszczególne przedmioty. Dział promocji i marketingu może być zainteresowany w otrzymaniu informacji o wynikach rekrutacji w kilku ostatnich latach na wybrane kierunki w zależności od miejsca zamieszkania (powiatu).

3. Zapytania do bazy danych

Aby prawidłowo zaprojektować bazę danych wspomagającą system zarządzania wyższą szkołą niepaństwową należy przeanalizować potrzeby informacyjne użytkowników informacji, czyli rozpoznać, jakie mogą być zapytania do bazy danych różnych grup użytkowników.

Zapytania kierownictwa wydziału (dziekana, prodzieka-

1Wydział Informatyki, Wrocławska Wyższa Szkoła Informatyki Stosowanej, ul. Wejherowska 28, 54-239 Wrocław, swietlana@lebiediewa.com

nów)
- Kto może prowadzić wykład z przedmiotu X?
- Kto może prowadzić ćwiczenia (laboratorium, projekt) z przedmiotu X?
- Jakie przedmioty może prowadzić wykładowca Y?
- Jakie jest pensum wykładowcy Y?
- Jakie przedmioty prowadzi wykładowca Y w semestrze *i*?

Zapytania pracowników dziekanatu:
- Czy student X (nr indeksu) może być wpisany na semestr (nr semestru)?
- Jakie są zaległości opłat studenta Y (nr indeksu)?
- Ile dyplomantów ma wykładowca X (nr wykładowcy)?
- Kto jest promotorem studenta Y (nr indeksu)?
- Jaka jest średnia ocen studenta Z (nr indeksu)?
- Jaki jest adres studenta X?

Wszystkie zapytania do bazy danych realizuje problemowy język zapytań.

4. Operacyjne i analityczne bazy danych w podejmowaniu decyzji

Zadaniem systemu informatycznego wspomagającego zarządzanie wyższą szkołą niepaństwową jest nie tylko dostarczanie potrzebnej informacji, ale w zależności od posiadanej informacji wspomaganie podejmowania decyzji. Zadaniem systemów wspomagających podejmowanie decyzji nie jest wyręczanie kierownictwa w podejmowaniu decyzji, lecz pomoc w jej opracowaniu [3]. Decyzje można podzielić na dwie grupy: decyzje operacyjne (decyzje do zarządzania szkołą „na co dzień") i decyzje strategiczne. Decyzje operacyjne są stosunkowo proste i dają się łatwo sformalizować i zaprogramować. Są to decyzje podejmowane w warunkach pewności. Znacznie trudniejsze lub niemożliwe do zaprogramowania są decyzje podejmowane przez kierownictwo szkół niepaństwowych w warunkach ryzyka i niepewności, są to decyzje strategiczne [4].

Operacyjna baza danych jest wykorzystana do zarządzania szkołą „na co dzień", czyli w dowolnym momencie. Operacyjna baza danych jest wykorzystywana do wspomagania pracy dziekanatu, administracji szkoły, planowania studiów, działu finansowego i służy do wspomagania decyzji w krótkim okresie czasowym. Operacyjna baza danych zawiera informację o zaliczeniach i egzaminach, o rozkładzie zajęć, o wykonaniu pensum przez poszczególnych wykładowców, o tym, jakie przedmioty *mogą prowadzić* poszczególni wykładowcy i jakie przedmioty *prowadzą*. Operacyjna baza danych wydaje informację „na bieżąco" i jest często aktualizowana (na przykład po każdym wniesieniu opłat przez studenta, po zaliczeniu semestru).

Natomiast analityczna baza danych dostarcza informacji do analizy problemu. Do analitycznej bazy danych należy informacja o miejscowościach, skąd pochodzą abiturienci, grupach wiekowych studentów szkoły, ich sytuacji materialnej, rodzinnej, o ich oczekiwaniach, a także informacja o absolwentach szkoły, ich zatrudnieniu, karierze zawodowej, o zakładach pracy, w których absolwenci znajdują zatrudnienie [5]. W odróżnieniu od operacyjnej bazy danych informacja należąca do analitycznej bazy danych ma charakter trwały, aktualizacja następuje rzadko. Dane należące do analitycznej bazy danych mają charakter historyczny. Analityczna baza danych służy do podejmowania decyzji strategicznych. Na przykład, badanie i analiza oczekiwań powoduje rozwój kierunków, powołanie specjalności oczekiwanych przez studentów. Z kolei powołanie nowych kierunków i specjalności wymaga podjęcia decyzji o zapewnieniu zasobów materialnych (sal wykładowych, laboratoriów) i kadrowych, a także zapewnienie miejsc w zakładach pracy do odbywania praktyk.

Przykładami decyzji operacyjnych są decyzje o dopuszczeniu studenta do sesji przeniesieniu studenta na następny semestr, przydzielenie zajęć wykładowcom. Aby podjąć decyzję o dopuszczeniu studenta do sesji, system zarządzania bazą danych sprawdza, czy jest uiszczone czesne, jeżeli tak, system drukuje kartę zaliczeń dla każdego studenta i listy studentów dopuszczonych do sesji w semestrze. W drugim przypadku, ponieważ warunkiem dokonania wpisu na następny semestr jest zaliczenie sesji i uiszczenie czesnego, system zarządzania baza danych sprawdza, czy student ma pozytywne oceny z egzaminów (zaliczeń) w sesji oraz ma uregulowane czesne. Jeżeli tak, student zostaje umieszczony na listę studentów wpisanych na odpowiedni semestr. Fragment bazy danych zawierającej informacje o czesnym i zaliczeniach przedstawiono na rys. 1, 2.

ZALICZENIA-EGZAMINY

Id-Stud	Id-Ku	Nr-Sem	Ocena
1001	Ba-Da-Ć	3	4
1001	Ba-Da-L	3	4
1001	Ba-Da-W	3	4
1003	Ba-Da-Ć	3	5
1003	Ba-Da-L	3	4
1003	Ba-Da-W	3	5
1006	Ba-Da-Ć	3	5
1006	Ba-Da-L	3	5
1006	Ba-Da-W	3	5
1007	Ba-Da-Ć	3	3
1007	Ba-Da-L	3	3
1007	Ba-Da-W	3	3

Rys.1. Tabela ZALICZENIA-EGZAMINY
Fig. 1. The table EXAMS-TESTS

CZESNE	Nr-Sem	Li_Rat	Zapłacone
1001	3	1	1
1003	3	2	1
1006	3	3	1
1007	3	2	2
1001	4	1	1
1003	4	2	1
1006	4	3	1
1007	4	2	0

Rys.2. Tabela CZESNE
Fig. 2. The table TUITION FEES

Poniżej przedstawiono dwa przykłady decyzji operacyjnych: decyzji o umieszczeniu na stronie internetowej listy studentów dopuszczonych do sesji w semestrze nr 3 i decyzji o wpisie studenta na następny semestr i umieszczeniu listy studentów wpisanych na następny semestr na stronie internetowej.
Zadanie 1: Umieść na stronie internetowej listę studentów dopuszczonych do sesji w 3 semestrze.
Założenie semantyczne: Warunkiem dopuszczenia do sesji jest uiszczenie czesnego.
Zdanie problemowego języka zapytań ma postać:
Umieść na stronie internetowej listę studentów dopuszczonych do sesji w semestrze nr 3.
Algorytm wykonania w języku algebry relacji:
Dla każdej krotki relacji CZESNE jeśli „Li-Rat" = „Zapłacone",
PROJECT ON Id-Stud. ⇒ LISTA STUDENTÓW drukuj *listę studentów.*
W rezultacie wykonania algorytmu otrzymujemy relację

LISTA STUDENTÓW

Id-Stud
1001
1007

Zadanie 2: Umieść na stronie internetowej listę studentów wpisanych na następny semestr.
Założenie semantyczne: warunkiem dokonania wpisu na następny semestr jest zaliczenie sesji i uiszczenie czesnego (w całości lub na raty).
Zdanie problemowego języka zapytań ma postać:
Umieść na stronie internetowej listę studentów wpisanych na semestr nr 4.
W celu wykonania algorytmu utworzymy perspektywę **ZALICZONY-3-SEMESTR** za pomocą instrukcji **CREATE VIEW:**
CREATE VIEW ZALICZONY-3-SEMESTR **AS**
SELECT Id-Stud
FROM ZALICZENIA-EGZAMINY
WHERE Nr-Sem=3 **AND** Ocena ≠ 2
W rezultacie wykonania algorytmu otrzymujemy relację

ZALICZONY-3-SEMESTR

Id-Stud
1001
1003
1006
1007

Następnie tworzymy perspektywę **ZAPŁACONY-4 SEMESTR:**
CREATE VIEW ZAPŁACONY-4-SEMESTR **AS**
SELECT Id-Stud
FROM CZESNE
WHERE Nr-Sem=4 **AND** Zapłacone≠0

Otrzymujemy relację

ZAPŁACONY-4-SEMESTR

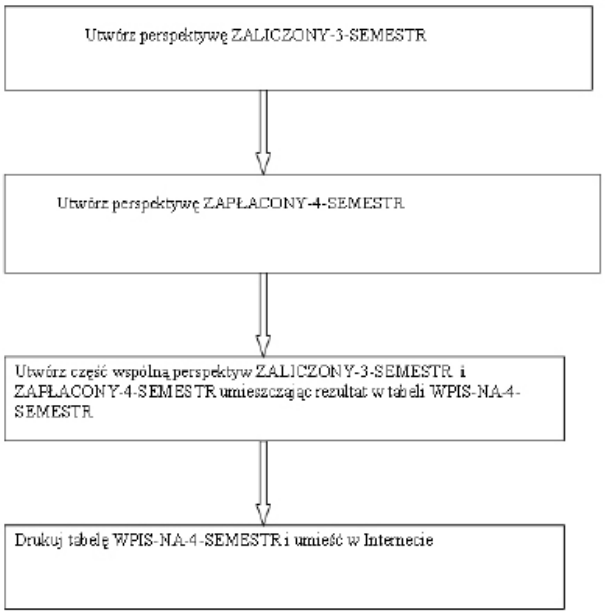
Id-Stud
1001
1003
1006

Wykonując operację **INTERSECT** (iloczyn teoriomnogościowy) na relacjach **ZALICZONY-3-SEMESTR i ZAPŁACONY-4-SEMESTR** otrzymujemy relację WPIS--NA-4-SEMESTR

WPIS-NA-4-SEMESTR

Id-Stud
1001
1003
1006

Relację **WPIS-NA-4-SEMESTR** drukujemy i umieszczamy w Internecie. Na rys. 3 przedstawiamy schemat blokowy algorytmu.



Rys. 3. Schemat blokowy algorytmu
Fig. 3. Flowchart of the algorithm

Bardziej złożona jest decyzja przydzielenia wykładowców dla zajęć laboratoryjnych dla kursu *k* w semestrze *j*. W celu wykonania zadania tego zadania trzeba wiedzieć, ile jest grup laboratoryjnych w semestrze *j* dla kursu *k*, utworzyć listy grup laboratoryjnych i listę wykładowców, prowadzących kurs *k*. Następnie należy sprawdzić, w jakim czasie (dzień, godzina) mogą odbyć się zajęcia kursu k dla wykładowców i grup, znaleźć wolne sale w tym czasie i przydzielić grupy poszczególnym wykładowcom, uwzględniając ich obciążenia. Decyzje dotyczące zarządzania procesem dydaktycznym korzystają z informacji operacyjnej zawartej w operacyjnej bazie danych [6].

Znacznie trudniejsze lub niemożliwe do zaprogramowania są decyzje podejmowane przez kierownictwo szkół niepaństwowych w warunkach ryzyka i niepewności. Są to decyzje strategiczne, do podejmowania których jest przydatna nie operacyjna lecz analityczna baza danych. Analityczna baza danych dostarcza informacji do analizy problemu. Do analitycznej bazy danych należy informacja o miejscowościach, skąd pochodzą abiturienci, grupach wiekowych studentów szkoły, ich sytuacji materialnej, rodzinnej, o ich oczekiwaniach, a także informacja o absolwentach szkoły, ich zatrudnieniu, karierze zawodowej, o zakładach pracy, w których absolwenci znajdują zatrudnienie. Badanie i analiza oczekiwań powoduje rozwój kierunków, powołanie specjalności oczekiwanych przez studentów. Z kolei powołanie nowych kierunków i specjalności wymaga podjęcia decyzji o zapewnieniu zasobów materialnych (sal wykładowych, laboratoriów) i kadrowych, a także zapewnienie miejsc w zakładach pracy do odbywania praktyk. Osoby podejmujące decyzje potrzebują zwykle informacji sumarycznych w różnych stopniach szczegółowości, analizy informacji sumarycznej w rozbiciu na wydziały, kierunki i specjalizacje, możliwości „wycięcia" informacji w zależności od zapotrzebowania, możliwości obejrzenia informacji na przestrzeni czasu, możliwości wyświetlenia informacji w postaci wykresów i tabel [7].

5. Prezentacja informacji

Istotnym problemem systemów informatycznych jest problem prezentacji informacji. Dla operacyjnych baz danych dogodnym środkiem prezentacji informacji są dwuwymiarowe tabele relacyjnej bazy danych otrzymane w procesie normalizacji. Przykłady zapytań do operacyjnej bazy danych:

1. Jakie przedmioty może prowadzić AAAA?
2. Jakie przedmioty prowadzi AAAA w semestrze zimowym?
3. Kto może prowadzić wykład z Baz danych i Algorytmów i struktur danych?
4. Kto może prowadzić zajęcia laboratoryjne z baz danych i systemów ekspertowych?
5. Kto jest dyplomantem AAAA?
6. Drukuj listę dyplomantów i tematy prac dyplomowych wykładowcy AAAA?

WYKŁADOWCY-PRZEDMIOTY

Wykładowca	Przedmiot
AAAA	Algorytmy i struktury danych
AAAA	Podstawy programowania
AAAA	Programowanie obiektowe
BBBB	Programowanie obiektowe
BBBB	Systemy operacyjne
CCCC	Bazy danych
CCCC	Hurtownie danych
CCCC	Algorytmy i struktury danych

PRZEDMIOTY- WYKŁADOWCY

Przedmiot	Wykładowca
Algorytmy i struktury danych	AAAA
Algorytmy i struktury danych	CCCC
Podstawy programowania	AAAA
Programowanie obiektowe	AAAA
Programowanie obiektowe	BBBB
Systemy operacyjne	CCCC
Bazy danych	CCCC
Hurtownie danych	CCCC

Rys. 4. Tabele WYKŁADOWCY-PRZEDMIOTY PRZEDMIOTY- WYKŁADOWCY
Fig. 4. The tables LECTURERS-COURSES and COURSES-LECTURERS

7. Ile stanowisk ma laboratorium L1?

Na rysunku Rys. 4. przedstawiono perspektywy WYKŁADOWCY-PRZEDMIOTY i PRZEDMIOTY- WYKŁADOWCY otrzymane z tabel bazowych relacyjnej bazy danych uczelni pozwalające odpowiedzieć na pytania 1 i 3. Na Rys. 5 i 6 przedstawiono perspektywy WYKŁADOWCY-DYPLOMANCI i SALE WYKŁADOWE-LABORATORIA pozwalające odpowiedzieć na pytania 5, 6 i 7.

WYKŁADOWCY-DYPLOMANCI

WYKŁDOWCA	STUDENT	TEMAT
AAAA	Nowak	Projektowanie sieci lokalnej
AAAA	Nowakowski	Projektowanie strony internetowej
CCCC	Kowal	Projektowanie internetowej bazy danych
CCCC	Kowalski	Projektowanie systemu informatycznego wspomagającego zarządzanie domem towarowym

Rys. 5. Tabela WYKŁADOWCY-DYPLOMANCI
Fig. 5. The table LECTURERS-GRADUATE STUDENT

SALE WYKŁADOWE-LABORATORIA

Numer sali	Rodzaj sali	Liczba miejsc/stanowisk
A1	Sala wykładowa	80
A2	Sala wykładowa	60
C11	laboratorium	18
C12	laboratorium	20

Rys. 6. Tabela SALE WYKŁADOWE-LABORATORIA
Fig. 6. The table LECTURE HALLS-LABORATORIES

Tabele w 3-ciej (i wyższych) postaciach normalnych zapewniają dogodną postać przechowywania informacji (brak redundancji, integralność danych) i zadowalającą obsługę transakcji. Jednocześnie takie struktury danych są mało przydatne do przeprowadzenia analizy zawartej w nich informacji. Analityczne bazy (hurtownie lub składnice danych) danych potrzebują innych struktur, specjalnego typu tablic decyzyjnych, specjalnych technik nazywanych *modelowaniem wielowymiarowym* i/lub specjalnych środków graficznych. Tablice decyzyjne (również modele wymiarowe) powstają na podstawie jednej lub kilku tablic wyjściowych bazy danych. Każdej kolumnie i każdemu wierszowi tablicy decyzyjnej odpowiada jakieś pole tablicy wyjściowej bazy danych. Dane na skrzyżowaniu wiersza i kolumny są otrzymywane w rezultacie zastosowania funkcji agregowania do pól tablicy wyjściowej. Tablice decyzyjne mogą być jednowymiarowe lub wielowymiarowe. Tablice jednowymiarowe są wykorzystywane do analizy zależności danych od jednego czynnika. Przykładem może być tablica zawierająca informację o liczbie zgłoszeń na studia w zależności od miejsca zamieszkania w jednym wybranym roku. Tablice wielowymiarowe są przedmiotem analizy wielowymiarowej. Kierownictwo szkoły jest zainteresowane badaniem zależności takich faktów jak rekrutacja, przyjęcia na studia, rezygnacja ze studiów, przeniesienie z innych szkół od następujących czynników (wymiarów) jak wydział (kierunek, specjalność), miejscowość opłata za studia, rok.

Przykłady zapytań do analitycznej bazy danych [8]:

- Jaka była liczba studentów na wydziale pedagogiki w 2004 r.?
- Jaki była ogólna liczba studentów w 2006 r.?
- Jakie 3 wydziały (specjalności) były najbardziej popularne w 2007 r. i jakie zaszły zmiany w stosunku do poprzednich dwóch lat?
- Jaki byłby wynik rekrutacji jeśli koszty o 3,5%, a czesne zostałyby obniżone o 10%?

Użytkownicy analitycznych baz danych zwykle potrzebują następujących informacji:

- informacji sumarycznych o różnym stopniu szczegółowości;
- analizy informacji sumarycznych według różnych jednostek organizacyjnych (wg wydziałów i miast);
- możliwości „wycięcia" informacji w wybrany sposób;
- możliwości wyświetlenia informacji w postaci graficznej i tabelarycznej;
- możliwości obejrzenia informacji na przestrzeni czasu.

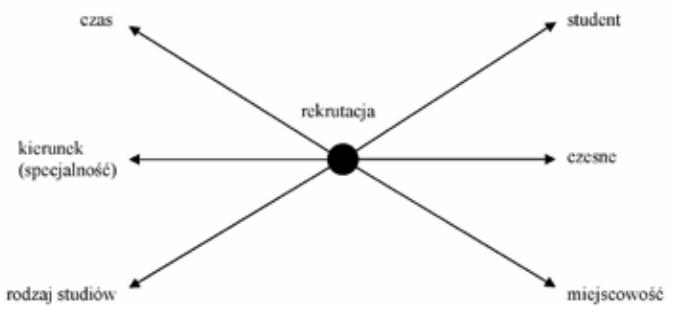
Raport z działalności szkoły za wybrany okres czasu może zawierać następującą informację (wymiarem jest *wydział*):

Nazwa wydziału,

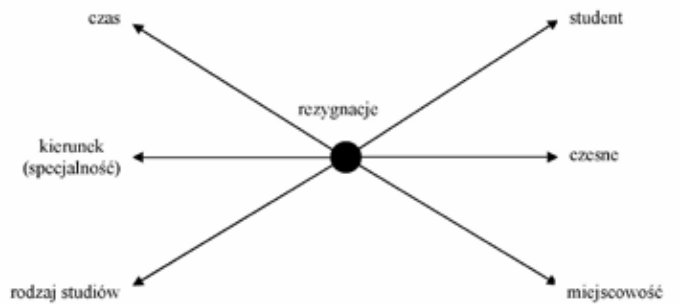
- liczba studentów,
- czesne,
- łączny dochód,
- łączny koszt,
- zysk brutto.

Wymiar może dotyczyć również *studenta*, *obszaru* i *czasu*.

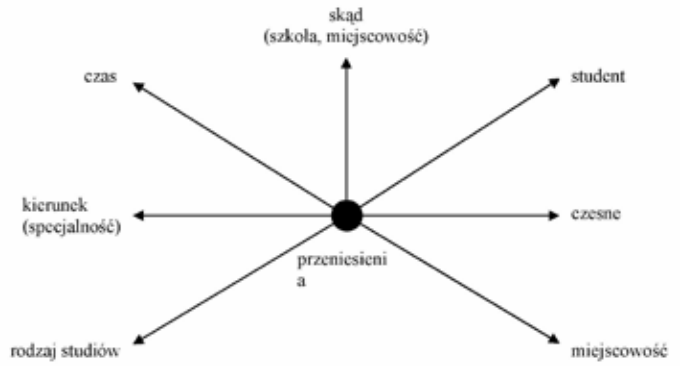
Poniżej przedstawiono zależność faktów przyjęcia na studia (rys. 7 - Rekrutacja), rezygnacja ze studiów (rys. 8 - Rezygnacje), przeniesienie z innych szkół (rys. 9 - Przeniesienia).



Rys. 7. Rekrutacja
Fig. 7. Recruitment



Rys. 8. Rezygnacje
Fig. 8. Resignations



Rys 9. Przeniesienia
Fig. 9. Transfers

6. Uwagi końcowe

Szkoła jak każdy podmiot gospodarczy ma swoje cele biznesowe. W przypadku szkoły celem biznesowym jest pozyskanie większej liczby studentów a także zwiększenie oferty (propozycja atrakcyjnych kierunków i specjalności, studiów podyplomowych).
Cele biznesowe przedsiębiorstwa mogą być osiągnięte tylko przy sprawnym zarządzaniu, które między innymi polega na podejmowaniu decyzji operacyjnych i strategicznych. Z kolei podejmowanie decyzji w dobie obecnej jest ściśle związane z informatyką w szczególności z systemami informatycznymi. Systemy informatyczne wspomagające zarządzanie operacyjne oparte są na operacyjnych bazach danych, które wspomagają podejmowanie decyzji na co dzień w każdej dziedzinie: proces dydaktyczny, zarządzanie kadrą, działalność finansowa. Systemy informatyczne wspomagające zarządzanie strategiczne oparte są na analitycznych bazach danych, które zawierają informacje archiwalne gromadzone przez szereg lat.

Literatura (References)

[1] Z. Biniek, Informatyka w Zarządzaniu, WIZJA PRESS & IT, Warszawa 2009.
[2] J. Kisielnicki, Systemy Informatyczne Zarządzania, PLACET, Warszawa 2008.
[3] A. Januszewski, Funkcjonalność informatycznych systemów zarządzania, t. 1 i 2 PWN, Warszawa 2008.
[4] A. Delorme, J. Jakymiw, S. Lebiediewa, Informatyka i socjologia w zarządzaniu szkołą wyższą W: Funkcjonowanie i Rozwój Organizacji w zmiennym otoczeniu III, Legnica 2003.
[5] R. Griffin, Podstawy zarządzania organizacjami, PWN, Warszawa, 2002.
[6] S. Lebiediewa, Rola bazy danych w zarządzaniu wyższą szkołą niepaństwową w: „Poszukiwanie modelu wyższej szkoły niepaństwowej" (2), Legnica 2004.
[7] C. Todman, Projektowanie hurtowni danych, Wydawnictwa Naukowo-Techniczne, Warszawa 2005.
[8] S. Lebiediewa, E. Dziubecki, Informatyka w zarządzaniu wyższą szkołą niepaństwową, w: Doskonalenie usług edukacyjnych w szkołach wyższych, Legnica 2009.

Application of Monte Carlo method in the computer system for valuation of exotic option contracts

Zastosowanie metody Monte Carlo w informatycznym systemie wyceny egzotycznych kontraktów opcyjnych

Hubert Zarzycki[1]

Treść. W artykule prezentowana jest propozycja komputerowego systemu obliczania finansowych instrumentów pochodnych – opcji egzotycznych i hybrydowych. Handel takimi produktami odbywa się na rynku pozagiełdowym (OTC) i często są to produkty tworzone na zlecenie. Wartości pewnych rodzajów opcji egzotycznych i hybrydowych nie można wyliczyć tradycyjnymi analitycznymi i numerycznymi metodami. W takich przypadkach warto użyć metody Monte-Carlo do wyceny rzeczywistej wartości instrumentów finansowych. W artykule przedstawiony został sposób obliczania ceny przykładowej opcji egzotycznej za pomocą metody MC. System komputerowy oparty o MC mógłby służyć do wspomagania decyzji inwestycyjnych dotyczących egzotycznych kontraktów opcyjnych.

Słowa kluczowe: metody Monte-Carlo, systemy wspomagania decyzji, opcje egzotyczne i hybrydowe, inżynieria finansowa, wycena kontraktów opcyjnych

Abstract. This paper presents a proposal for a computer system of calculation of financial derivatives - exotic and hybrid options. The trade of such products takes place on the OTC market and these products are often made and tailored on demand. The values of certain types of exotic and hybrid options cannot be calculated with traditional analytical and numerical methods. In such cases, it is worth to use the Monte-Carlo method for the valuation of the real value of financial instruments. This paper presents an example of calculating the price of exotic options using the MC method. The computer system based on MC could be used to support investment decisions regarding exotic option contracts.

Keywords: Monte-Carlo methods, decision support systems, exotic and hybrid options, financial engineering, valuation of option contracts

1. Introduction

Since the second half of the twentieth century the rapid development of applications for option contracts has been noted. These products make it possible to achieve above-average investment returns regardless of economic conditions. They can be based on various underlying assets such as stocks, currencies, indices and commodities. According to the theory [11, 14], an option is a contract between the buyer and a seller that gives the buyer the right (but not the obligation) to call or put underlying asset at a specified time in the future at a preset price, in exchange for a fee called the option premium. The option price is dependent on the value of the underlying (base) assets. Having the option gives you the right, but no obligation, and therefore the option holder can exercise this right, when it is profitable for him. The exercise of the right is called the exercise of option or its settlement. The holder of the American option exercises it at any time from the moment of purchase to the date of termination. European option can only be exercised on the option's expiry date. So in the case of the European option the exercise and expiration dates are the same. American and European types of option contracts are referred to as standard or vanilla ones. Any other option contracts are called exotic or hybrid.

Financial institutions seeking more efficient and effective management methods of capital in the financial markets began to commonly use option contracts with the arrival of the first analytical [1] and discrete [6] models of the evaluation of these financial products. There are many reasons why exotic and hybrid options are attractive to investors. Among the most important, one may mention that the usual price of exotic and hybrid options is lower than the one of the standard options, and in most cases they give similar and adequate investment and security as vanilla options (American and European). Besides, new contracts created also expand risk management capabilities. However, technological development, reflected by the increase in computing power, can solve complex numerical problems related to the valuation of financial products in real time.

Larger and more diverse requirements of investors lead to the formation of successive derivatives. Among the recently introduced contracts, significant and increasingly important role is played by exotic and hybrid options. These are products of a more complex structure than the standard options. Trading in these instruments can be done at Over The Counter market (OTC) and/or they can be tailored to a specific investor's needs. In the case of valuation of ta-

1Wydział Informatyki, Wrocławska Wyższa Szkoła Informatyki Stosowanej, ul. Wejherowska 28, 54-239 Wrocław, hzarzycki@horyzont.eu

ilored instruments, frequently important is the efficiency of the preparation of a valuation model [5, 16]. Analytical methods cannot be used to describe the valuation formula of all the exotic option contracts. And given that certain types of exotic options and hybrids cannot be measured by any traditional methods, it makes sense in such cases to use the Monte Carlo simulation to evaluate the price of the financial instruments concerned.

2. Monte Carlo method for valuation of option contracts

In this chapter the effective MC method [8, 16] of valuation of exotic options with a sample valuation of a simple Lookback type contract will be presented. Monte Carlo method may be used in a variety of stochastic processes, even if the natural logarithm of the value of the basic instrument behaves in accordance with the geometric Brownian motion:

$$dS=\left(1+\mu S+\frac{1}{2}\sigma^2 S\right)dt+\sigma S dZ$$

Using Monte Carlo methods for the valuation of options was first presented by Boyle [2, 10, 11]. MC is usually used when there are no analytical and discrete models. That is, when the price of the basic instrument S is given by:

$$S+\Delta S = Se^{(\mu-\sigma^2/2)\Delta t+\sigma\varepsilon_t\sqrt{\Delta t}}$$

where dZ is the differential of the Wiener process [4, 12], with a standard deviation σ equal to one and the average value (drift) μ equal to zero. In order to simulate the process, we split the timeline into a finite number of n intervals of the following ends $t_0, t_1, t_2, ..., t_n$ away from each other of Δt symbol,

$$S+\Delta S = Se^{(\mu-\sigma^2/2)\Delta t+\sigma\varepsilon_t\sqrt{\Delta t}}$$

where ΔS is the growth rate of S price in a given period of time Δt, and ε_t are independent random values generated from the standard normal distribution $N(0,1)$. Most programming languages have built-in function that returns a random value from a normal distribution. If, however, there are only random Z numbers from 0 to 1, they have to be converted into a random value of the normal distribution. Subsequent independent S_n values are calculated using the following rule [10]:

, for m=1…n

$$S_m = S_{m-1} e^{(\mu-\sigma^2/2)(t_m-t_{m-1})+\sigma\varepsilon_t\sqrt{t_m-t_{m-1}}}$$

Assuming that the payoff function of the W option depends only on the value of the basic S_n one can find the formula for the option price C for a single trajectory (Figure 1) of the basic instrument prices $S_0, S_1, S_2, ..., S_n$:

$$C = e^{-rT} W(S_0, S_1, S_2, \ldots, S_n)$$

but after the performance of k experiments (Figure 2), the option premium based on average is equal to:

$$C = e^{-rT}\frac{1}{k}\sum_{eks=1}^{k} W\left(S_{t0}^{eks}, S_{t1}^{eks}, S_{t2}^{eks}, \ldots, S_{tn}^{eks}\right)$$

It needs to be borne in mind that for each type of option the payment function W is calculated differently. Each of the trajectories in Figure 2 corresponds to one experiment from the formula.

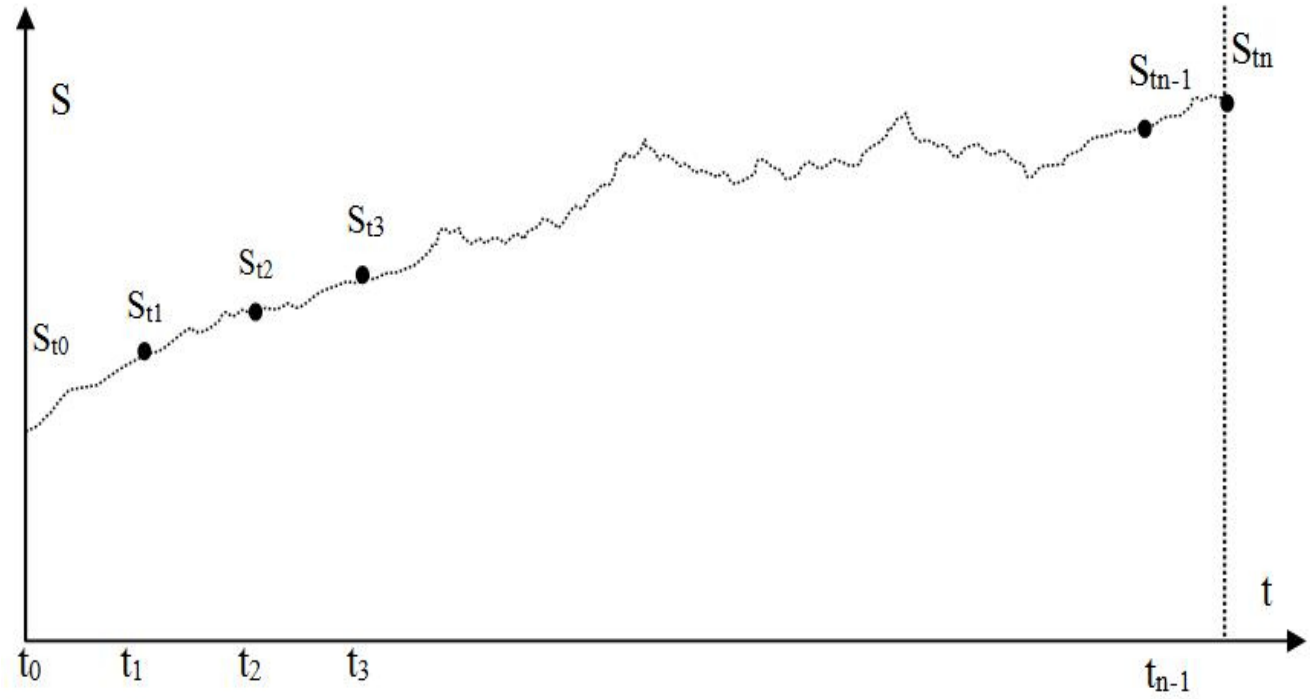


Fig. 1. A single price trajectory of underlying instrument S.
Rys. 1. Pojedyncza trajektoria cen S instrumentu podstawowego.

Source: own research

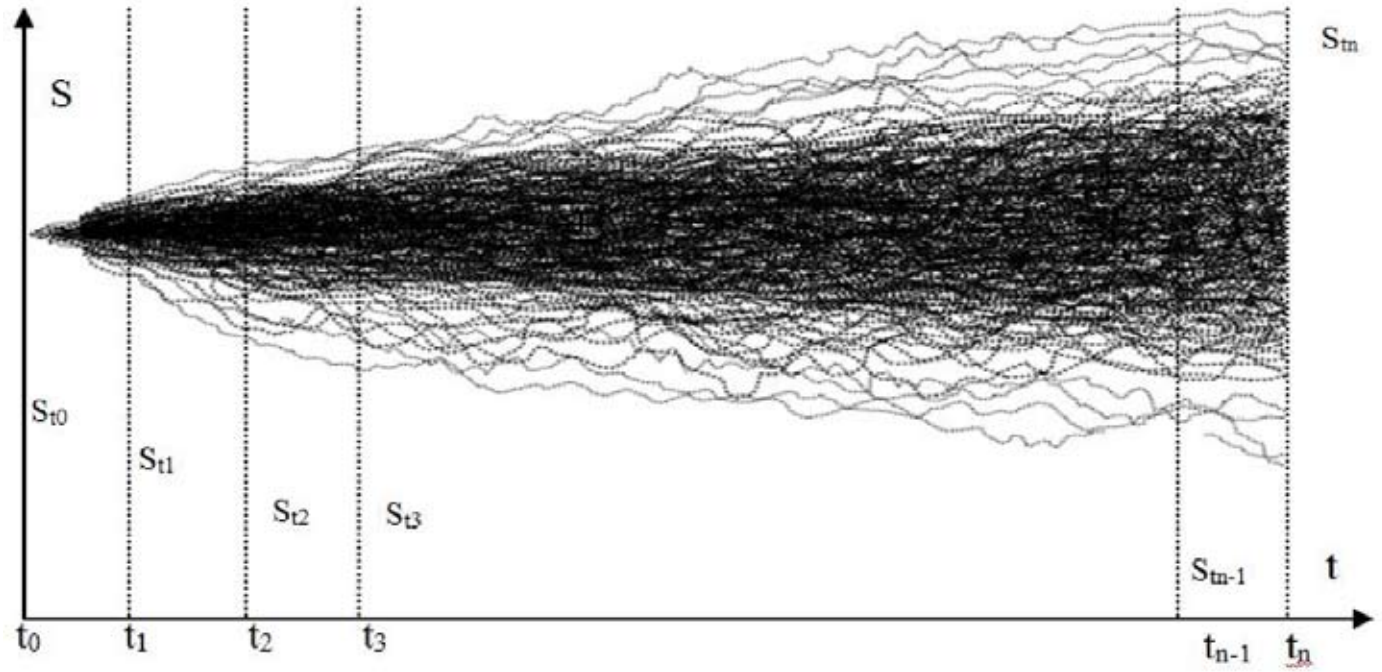


Fig. 2. Random trajectories of the underlying asset price S after performance of k experiments.
Rys. 2. Losowe trajektorie cen instrumentu bazowego S po przeprowadzeniu k eksperymentów.
Source: own research, [17]

Classic method of valuation, proposed in the original version, using the Monte Carlo simulation is relatively ineffective. Satisfactory option premium valuation accuracy is achieved only after generating at least ten thousand *k* trajectories. In addition, for up to 10-fold increase in the accuracy of calculations it is required to carry out up to 100 times more experiments [8]. A number of techniques to speed up the described method was developed. The most often used methods are applied via the reduction of variance [3, 11]. These include:

- antithetic variates
- stratified sampling
- importance sampling
- control variates
- moment matching.

One of the most practical technique is the antithetic variates method which consists for every sample (primary) path obtained additional symmetric antithetic path. This technique has two advantages. It decreases the variance of the sample paths, improving the accuracy. Also it reduces the number of samples to be generated to obtain N paths. Figure 3 below shows exemplary paths.

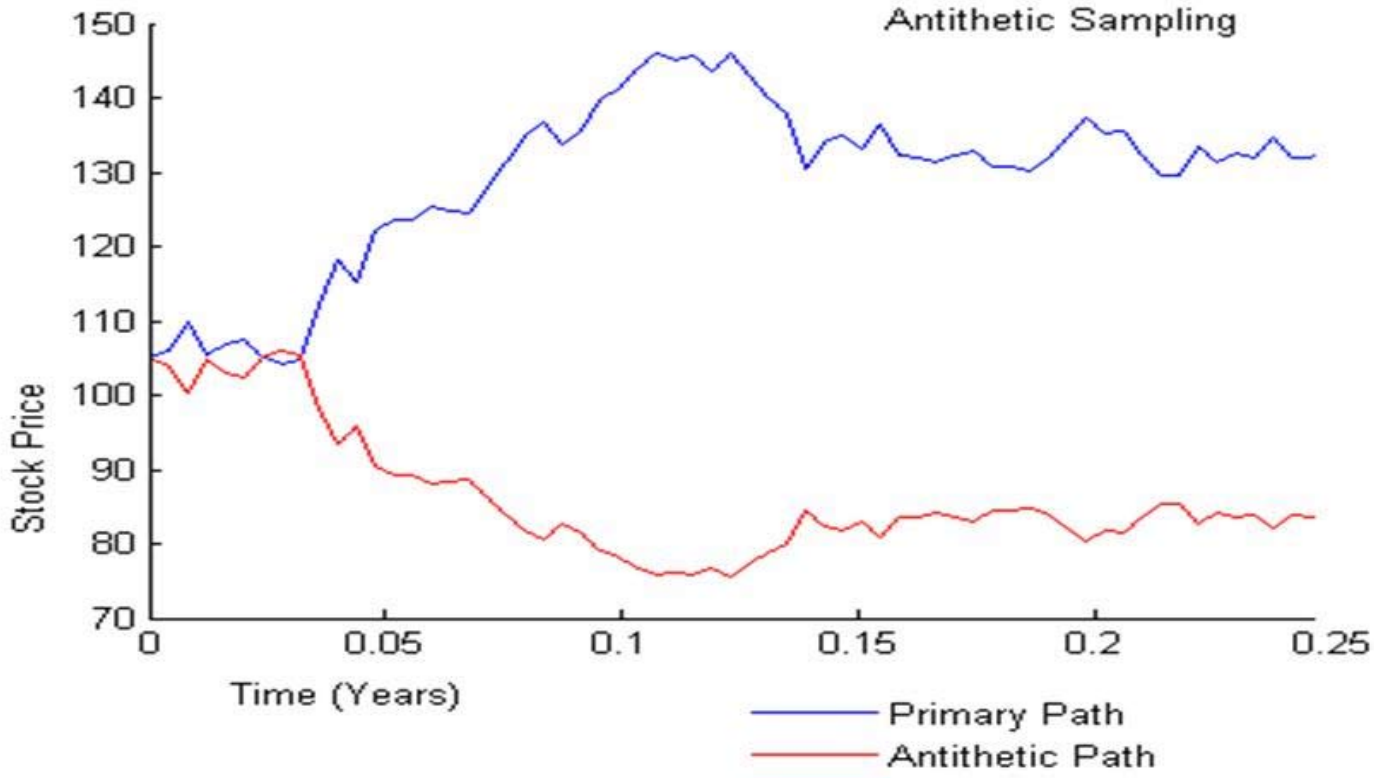


Fig. 3. Symmetric primary and antithetic sample paths.
Rys. 3. Symetryczne ścieżki próbki pierwotnej i antytetycznej.
Source: [18], own research

3. Applying the MC for the Lookback option valuation

We will now consider an effective method of valuation using Monte Carlo method for Lookback option – popular exotic contract. There are two basic types of options; call and put. The value of the payment from option depends upon the exercise price *X*. The payment function for the Lookback contract is also subject to the average S_A price of the underlying instrument. The following table shows the functions of payment *W* of Lookback options.

Tab. 1. Payment functions of Lookback options.
Tab. 1. Funkcje wypłaty opcji wstecznych.

Type:	Payment function
Lookback floating-strike call	$W_{call} = \max\left(0, S(T_{Final}) - \min_{t_i} S(t_i)\right)$
Lookback floating-strike put	$W_{put} = \max\left(0, \min_{t_i} S(t_i) - S(T_{Final})\right)$

Source: own research

According to the approach proposed by Broadie and Glasserman [3] in order to perform a single experiment, one may needs to find values at t_i. The examples below shows the formula of calculating the value of the option at expiry (payoff function) using MC method:
where:

T_0 – option start date
T_{Final} – final payment date

$$W_{call} = \max\left(0, \frac{S(T_{Final}) - \min_{t_i} S(t_i)}{S(T_0)}\right)$$

$$W_{put} = \max\left(0, \frac{\max_{t_i} S(t_i) - S(T_{Final})}{S(T_0)}\right)$$

$t_1, t_2, \ldots, t_m$ with $T_0 \le t_1 < t_2 < \ldots < t_m \le T_{Final}$ - Lookback option dates
S(t) denotes the price of the underlying at time t.
After performing *k* experiments and discounting W_{call} (W_{put}) to the current value, one can obtain option price:

$$C_{call} = e^{-iT} \frac{1}{k} \sum_{eks=1}^{k} W_{call}^{eks}$$

$$C_{put} = e^{-iT} \frac{1}{k} \sum_{eks=1}^{k} W_{put}^{eks}$$

When setting the value of the option, usually the price sensitivity factors are calculated at the change in the factors that determine the price. The factors that affect the price of an option are: the price of the underlying instrument *S,* exercise price *X,* volatility of the underlying instrument σ, the length of time for the expiration date *t,* and the risk-free interest rate *r.* Accordingly delta, gamma, lambda, theta and ro coefficients indicate how the option price will change when the factors affecting the price of an option will change the unit of measurement. Sensitivity coefficients are very valuable information, from an investor's perspective. Therefore, they are an essential part of the system of valuation. Detailed description of methodology for determination of the sensitivity coefficients in the MC method is beyond the scope of this paper. More on this topic can be found in the development by Michael C. Fu and Jian Qiang Hu [9].
The exemplary valuation results are presented below. N is the number of simulations. There is a minimum difference in results when N=1000 and N=5000. It shows that the method performs well comparing to Internet option calculators. It could be used for valuation of real market options.

Tab. 2. European lookback call. Parameters: S=100, X=100, σ=0,2, t=0,5, r=0,03.
Tab. 2. Europejska wsteczna opcja kupna Parametry S=100, X=100,σ=0,2, t=0,5, r=0,03.

N	Option value
100	9.251
500	9.524
1000	9.653
5000	9.701

4. The computer system for evaluating the value of exotic options

According to Turban, decision support systems [15, 7, 13, 16] are designed for decision-makers at various levels of the organization and deal with the semi-structuralized decisions. A financial analyst meets such problem during the decision making process, whether the timing is good to invest and call/put option contracts. Automated data processing, leading to a clear investment decision is not possible. If the method were possible to be expresses as an algorithm, the decision-making process would be structuralized and could be solved automatically. However, we deal with the problem of poor or poorly structuralized, in which there is no algorithm clearly indicating the correct choice. Only through guidance such as the theoretical va-

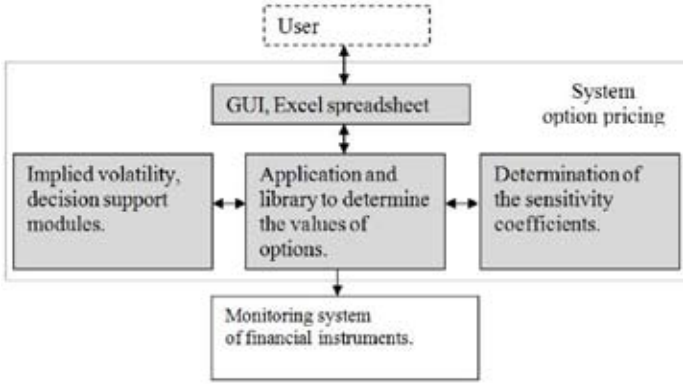


Fig. 4. Schematic diagram of the exotic options valuation system.
Rys. 4. Schemat poglądowy systemu wyceny opcji egzotycznych.
Source: own research

lue of the option and the sensitivity coefficients one can support decision-making processes.
The system design should focus both on improving the currently available solutions and larger offer of available instruments. Schematic diagram of the system is shown in Figure 4.
The data is entered into the system by the user into a spreadsheet forming a graphical user interface. It is worth mentioning that, spreadsheets, and especially Excel, are standard among financial analysts. GUI in the sheet should be based on an intuitive approach and prevent invalid data entry. For example, entry values for the options with unrealistic data such as incorrect dates must be rejected. After filtration and preparation of the input data with VB code, they are sent from the spreadsheet to the evaluation application. The designated value of the option can then be used to calculate the sensitivity coefficients. Additional functionality of the system, not described in this document due to the extensiveness, should include: determining implied volatility [3, 11], decision support modules, and preparation of charts of payoff function. All results are sent to the spreadsheet. The theoretical value of financial instruments and the sensitivity coefficients can be transferred to an external real-time system controlling open positions in the contracts.
Decision support module role is to indicate potential investment opportunities. For instance compare the market price of the option contract with the calculated theoretical price, track changes in the sensitivity coefficients (delta, gamma, theta, lambda, rho). Such information helps the investor to decide whether to buy or sell options.
Currently, in the software market, one can find a lot of tools that allow easy and rapid application development (eg, .Net). Application for evaluation of the options and the sensitivity coefficients should be formed by the fastest object oriented programming languages and also the fourth-generation languages (such as C++, C#). Their biggest advantage is the ability to create complex libraries without writing too much code in an automated way, based on standard components and features.

5. Conclusions

In order to create a practical system for the valuation of exotic options one must meet the high requirements set by the investors, especially regarding the speed and accuracy of calculations. Technological advances and the growing capabilities of computers allow the use of real-time pricing scheme for valuing options contracts based on the Monte Carlo methods. Presented valuation system is designed in such a way as to ensure the proper evaluation of financial products and enable efficient decision support for investment in both market and OTC options.
The biggest advantage of this system is to provide an uncomplicated prospect of adding further complex financial products algorithms. In most cases, for the introduction of a new product it will only be necessary to prepare and implement contract payment function. The payment function can use a common MC generator and associated libraries. A wide range of financial instruments available in the library will meet the growing demands of investors. The system should also have security mechanisms against the introduction of incorrect data, calculate implied volatility, sensitivity coefficients and have a decision support module. The application on the Polish market due to its abilities and novelty could be a practical and unique tool supporting various investment in options.

Literature (Literatura)

[1] F. Black, M. Scholes, The valuation of option contracts and a test of market efficiency, Journal of Finance, 1972 May, s 399-417.
[2] P. Boyle, Options: A Monte Carlo Approach, Journal of Financial Economics, 4, 1977, s. 323-338.
[3] M. Broadie, P. Glasserman, Monte Carlo methods for security pricing, Journal of Economic Dynamics and Control 21, s. 1267–1321.
[4] A. Cerny, *Mathematical Techniques in Financ*e, Princeton University Press, 2004.
[5] D. Chorafas, *Financial models and simulation,* McMillan Press, London 1995.
[6] J. Cox, S. Ross, M. Rubinstein, *Option pricing. A simplified Approach*. Journal of Financial Economics, 1979 March.
[7] J. Czerniński, *Systemy wspomagania decyzji w zarządzaniu przedsiębiorstwem*, TNOiK, Toruń 2002.
[8] D. Duffie, P. Glynn, *Efficient Monte Carlo estimation of security price*s, Ann. Applied Probability 5, 1996.
[9] M. Fu, J.Q. Hu, *Sensitivity analysis for Monte Carlo simulation of option pricing.* Probability in the Engineering and Information Sciences Vol 9, s. 417-446, Sept.1994, revised June 2005.
[10] E. Haug, *The complete guide to option pricing formulas*, McGraw Hill, 1998.
[11] J.C. Hull, Options. *Futures and Other Derivatives (Upper Saddle River, NJ)*, 6th Edition, Prentice Hall 2006.
[12] J. Jakubowski, A. Palczewski, M. Rutkowski, Ł. Stettner, *Matematyka finansowa*, Wydawnictwa Naukowo-Techniczne 2003 .
[13] E. Radosiński, *Systemy informatyczne w dynamicznej analizie decyzyjnej*. PWN 2001.
[14] W. Tarczyński, M. Zwolankowski, *Inżynieria Finansowa*, Agencja Wydawnicza PLACET, Warszawa 1999.
[15] E. Turban, *Decision Support Systems and Expert Systems*, Prentice Hall, Englewood Cliffs 1995.
[16] H. Zarzycki, *System ewaluacji kontraktów opcyjnych metodą Monte Carlo*, PSZW 2009.
[17] H. Zarzycki, *Wspomaganie decyzji w informatycznym systemie zarządzania inwestycjami na rynkach terminowych*, Praca doktorska, Politechnika Szczecińska, Wydział Informatyki 2006.
[18] http://www.mathworks.com/help/econ/simulate.html [ostatni dostęp: 14.05.2013].

Nowa implementacja algorytmu mrówkowego wykorzystująca technologię przetwarzania wieloprocesorowego i rozproszonego w systemie nawigacji

A new implementation of an ant algorithm using multiprocessor and distributed computing technologies in navigation system

Daniel Komar[1]

Treść. Artykuł ma na celu przybliżenie czytelnikowi problemu wyboru najlepszej trasy podróży pomiędzy dwoma punktami, która będzie minimalizowała liczbę negatywnych czynników wpływających na osobę kierującą pojazdem. Zaprezentowany zostanie nowo zaimplementowany algorytm mrówkowy, który został przystosowany do wykorzystania w pełni możliwości obliczeniowych współczesnych systemów wieloprocesorowych i rozproszonych. Autor przeprowadzając eksperyment w warunkach rzeczywistych, ukaże wyższość opracowanego rozwiązania nad stosowanym obecnie tradycyjnym systemem nawigacji. Przeprowadzone badania wykazały, że wykorzystywana nowa implementacja algorytmu w znacznym stopniu zmniejsza czas przejazdu i liczbę czynników zakłócających mających bezpośredni wpływ na osobę kierującą pojazdem.

Słowa kluczowe: algorytm mrówkowy, nawigacja, czas podróży, warunki drogowe

Abstract. The purpose of this paper is to give reader an understanding of the problem of the best itinerary selection between two points which will minimize the number of negative factors affecting the person driving a vehicle. The author will present an entirely new implementation of an ant algorithm that was adapted in order to make the most of computational capabilities of modern multiprocessor and distributed systems. Having performed experiments in real-world conditions, the author demonstrates, that the new solution is superior to the traditional navigation system which is still used today. The conducted research showed that the new implementation of the algorithm significantly contributed to reduction of journey times and the number of confounding factors which have a direct impact on the person driving a vehicle.

Keywords: ant algorithm, navigation, journey times, driving conditio

1. Wstęp

W ostatnich latach nastąpiło znaczne przyśpieszenie procesu urbanizacji, powodującego znaczny wzrost zaludnienia miast. Efektem tego procesu jest bardzo duży wzrost liczby pojazdów uczestniczących w ruchu drogowym, co powoduje powstawanie korków drogowych. Kierowcy, co prawda coraz częściej wykorzystują system nawigacji satelitarnej, który często nie jest doskonały, ponieważ nie uwzględnia zmieniających się warunków drogowych na wyznaczonej trasie. Problem wyznaczania nieoptymalnego rozwiązania względem panujących warunków powoduje powstanie związku przyczynowo skutkowego o znacznych rozmiarach. Głównym efektem tego jest negatywny wpływ na sprawność psychofizyczną kierowcy. Przekłada się to na spadek sprawności psychicznej i powoduje błędną percepcję osoby prowadzącej pojazd, co może prowadzić do podejmowania wielu pochopnych i błędnych decyzji w istotnych dla bezpieczeństwa sytuacjach drogowych.
W artykule zostanie przedstawiony algorytm mrówkowy zastosowany w systemie nawigacji, który został zaadaptowany do wyznaczania optymalnej trasy podróży z uwzględnieniem wielu istotnych parametrów zmieniających się w czasie rzeczywistym. W związku z tym podczas każdorazowego generowania nowego kierunku jazdy pomiędzy dwoma punktami na mapie algorytm stara się dostarczyć jak najlepsze rozwiązanie problemu. Zostaną również przedstawione czynniki, które zostały sklasyfikowane jako odgrywające dużą rolę w ruchu pojazdów.
W dalszej części zostanie zaprezentowany eksperyment przeprowadzony w warunkach rzeczywistych. Przeprowadzona próba była zrealizowana z udziałem dwóch niezależnych systemów nawigacji. Pierwszy wykorzystujący tradycyjne i obecnie stosowane na szeroką skalę rozwiązania oraz drugi wykorzystujący algorytm mrówkowy z uwzględnieniem wielu relewantnych czynników. Przeprowadzone badania mają na celu zbadanie możliwości usprawnienia komunikacji w dużych ośrodkach miejskich.

2. Tło koncepcji

Inspiracją do rozpoczęcia projektu systemu nawigacji wykorzystującego algorytm mrówkowy z uwzględnieniem aktualnie panujących warunków drogowych były psycho-

1Wydział Informatyki, Wrocławska Wyższa Szkoła Informatyki Stosowanej, ul. Wejherowska 28, 54-239 Wrocław, gordon1x@poczta.fm

logiczne badania kierowców prowadzone przez Zakład Psychologii Transportu Drogowego Instytutu Transportu Samochodowego w Warszawie. Analizowano w nich występujące w ruchu drogowym zakłócające czynniki mające zasadniczy wpływ na kierującego pojazdem. Przyjmując definicję bezpiecznego kierowcy, który ma na celu przeprowadzić bezkolizyjnie pojazd pomiędzy dwoma punktami (początkowym i docelowym) [2]. Kierowca nie mogąc przewidzieć splotu wszystkich czynników może natrafić na bodziec powodujący dezorganizację zamierzonych działań. Zaburzenia takie jak korek drogowy, utrudnienia w ruchu powodują u osób kierujących występowanie indywidualnych stanów emocjonalnych. Bardzo negatywny skutek na kierowcę mają silne emocje, spowodowane zbyt długo występującym zaburzeniem. Wpływają one w znaczący sposób na osłabienie sprawności psychofizycznej. Mocno niepożądanym skutkiem tego może być obniżenie szybkości reakcji, podzielności uwagi i koncentracji. Rezultatem tego jest znaczna zmiana zachowania kierowcy, który chce jak najszybciej osiągnąć cel swojej podróży. Zapomina o bezpieczeństwie i wykonuje w wielu przypadkach ryzykowne manewry. Podejmowane przez niego decyzje przestają być racjonalne, co naraża innych uczestników ruchu na kolizję.

Duży wpływ na wybór algorytmu mrówkowego do realizacji zadania nawigacji miały artykuły naukowe i badania przeprowadzone przez Marco Dorigo. W jego pracach można odnaleźć główne zasady jakie spełniać musi algorytm mrówkowy, aby umożliwić rozwiązanie problemu wyszukiwania najkrótszej trasy pomiędzy dwoma punktami [1, 3, 4, 5]. Przedstawione w pracach różne modyfikacje algorytmu wykorzystują podejście probabilistyczne, a w implementacji programowej generator liczb losowych.. Powoduje to, że znalezione rozwiązanie niekoniecznie jest najlepsze, ale w stosunku do innych algorytmów istnieje prawdopodobieństwo znalezienia zadowalającego rozwiązania w krótszym czasie.

3. Algorytm mrówkowy

Na potrzeby projektu została opracowana implementacja algorytmu mrówkowego, która była wzorowana na głównych zasadach funkcjonowania zawartych w pracach i publikacjach naukowych Marco Dorigo [3, 4].

Główną ideą algorytmu mrówkowego jest wykorzystanie zasad samoorganizacji występujących w naturalnym ekosystemie kolonii mrówek. Zastosowane zasady mają na celu koordynacje ich sztucznych, cyfrowych odpowiedników, które umożliwią rozwiązywanie problemów [4]. Inspiracją były różne zachowania mrówek: żerowanie, podział pracy, sortowanie i transport kooperacyjny. Biologowie prowadzący badania wykazali, że mrówki koordynują swoją pracę za pomocą pośredniego mechanizmu porozumiewania przy wykorzystaniu zmian środowiska zwanym stygmergią. Najlepsze rezultaty osiągnięto przy opracowaniu algorytmu bazującego na mechanizmie występującym podczas żerowania i jest on ukierunkowany na rozwiązywanie problemu optymalizacji kombinatorycznej [4].

Za główny punkt nowej implementacji przyjęto przystosowanie algorytmu do wykorzystania możliwości nowoczesnych systemów wieloprocesorowych, przetwarzania równoległego i rozproszonego. Dzięki temu rozwiązaniu będzie możliwe znaczne przyśpieszenie wykonywanych operacji. Pozwoli to wykorzystać moc obliczeniową procesorów wielordzeniowych, systemów wieloprocesorowych oraz klastrów obliczeniowych.

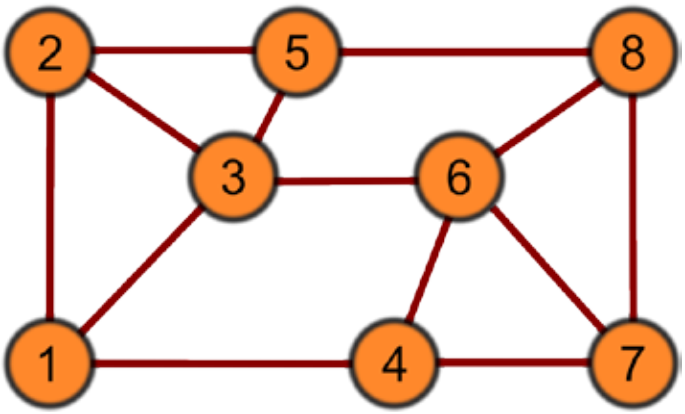
Zaimplementowany algorytm w procesie decyzyjnym uwzględnia również najważniejsze czynniki takie jak: odległość pomiędzy wyznaczonymi węzłami komunikacyjnymi, klasa drogi, maksymalna prędkość pojazdu, występowanie korka drogowego, utrudnienia drogowe. Należy podkreślić, że uwzględniany jest również rodzaj węzła komunikacyjnego takiego, jak skrzyżowanie równorzędne, o ruchu okrężnym, czy kierowane znakami lub sygnalizacją świetlną. Zależnie od jego rodzaju odpowiednio uwzględniono średni czas oczekiwania osoby prowadzącej pojazd na przejazd przez dany węzeł (skrzyżowanie). Należy zauważyć, że wszystkim uwzględnianym parametrom zostały przypisane odpowiednie wagi. Zdefiniowane we wcześniejszym etapie wartości wag poszczególnych parametrów pozwalają obliczyć wagę na odcinku zlokalizowanym pomiędzy dwoma indywidualnymi węzłami. Tym samym staje się możliwe wyznaczenie wag pomiędzy wszystkimi połączonymi parami węzłów zaznaczonych na fragmencie mapy.

Z założenia algorytm nie będzie wyznaczał statycznej trasy pomiędzy punktem startowym a docelowym, lecz będzie w czasie jazdy aktualizował wyznaczoną trasę wykorzystując zmieniające się dane o utrudnieniach występujących w ruchu drogowym. Przy wyborze trasy przejazdu algorytm szacuje przewidywany czas podróży. Jeżeli jego wartość będzie mniejsza to algorytm wyznaczy nową lepszą trasę podróży. Warto zauważyć, że uaktualnienie trasy wiąże się również z nowym oszacowaniem czasu podróży z węzła komunikacyjnego, w którym się w danej chwili pojazd znajduje, a punktem docelowym. Należy zwrócić uwagę, że obliczenia są wykonywane tylko i wyłącznie podczas pokonywania trasy pomiędzy węzłami w ściśle określonym czasie. Algorytm wyznacza nową trasę, kierowca zauważa informację, podejmuje decyzję i reaguje przed dojazdem do węzła komunikacyjnego.

4. Ogólna struktura algorytmu

Zanim zostanie przedstawiona ogólna struktura algorytmu należy zdefiniować pojęcie grafu. Graf jest to para złożona z dwu zbiorów: wierzchołków V(G) i krawędzi E(G). W grafie prostym krawędzie te są nieuporządkowanymi parami wierzchołków, natomiast w grafie skierowanym są uporządkowanymi parami wierzchołków [6]. W przypadku opisywanego problemu V zawiera zbiór węzłów komunikacyjnych, natomiast E zawiera zbiór wag obliczonych na podstawie charakterystycznych wartości parametrów

dla ruchu drogowego. Na potrzeby projektu wykorzystano strukturę grafu nieskierowanego (rys. 1.1.),



Rys. 1.1. Przykładowa graf dla algorytmu mrówkowego
Fig. 1.1. Sample the graph for ant algorithm

ze względu na możliwość optymalizacji złożoności wykonywanych obliczeń. Należy pamiętać, że muszą zostać spełnione ściśle określone założenia:

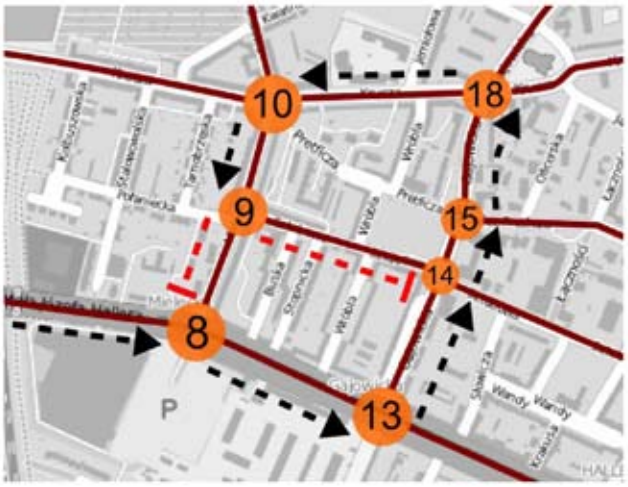
(1) W grafie G występuje unikalny węzeł, który jest oznaczony jako startowy.
(2) W grafie G występuje unikalny węzeł, który jest oznaczony jako końcowy.
(3) Niech T będzie zbiorem węzłów ścieżki t pomiędzy dwoma wyznaczonymi węzłami (1), (2) w grafie G spełniając następujące założenia:

1. Pierwszy element ścieżki t jest węzłem oznaczonym jako startowy w grafie G (1).
2. Ostatni element ścieżki t jest węzłem oznaczonym jako końcowy w grafie G (2).
3. W ścieżce t każdy z węzłów grafu G może wystąpić tylko raz.
4. Ścieżka t niekoniecznie zawiera wszystkie węzły znajdujące się w grafie G.
5. Elementy ścieżki t należą do liczb naturalnych z zerem włącznie i tym samym jest spełnione założenie:

$$T=\{t:t\in\mathbb{N}_0\}$$

List. 1.1. Uproszczony pseudokod algorytmu w języku programowania Pidgin ALGOL
List. 1.1. The simplified pseudocode of the algorithm in Pidgin ALGOL programming language

```
begin
        comment Wczytywanie danych i parametrów działania programu:
        comment I - liczba iteracji, M - wagi pomiędzy węzłami komunikacyjnymi
        comment S - obecny węzeł komunikacyjny, K - końcowy węzeł komunikacyjny
        comment P - liczba procesów
        read M, I, S, K, P

        comment Zmienne, do której mają dostęp wszystkie procesy programu
        koszt ← Ø
        trasa ← Ø

        for i ← 0 step 1 until (P - 1) do
            beginPARA
                    comment Funkcja generuje macierz
                    feromon ← GenerujMacierz(M)

                    for m  ← I step -1 until 0 do
                            comment Funkcje generujące trasę podróży
                            temp  ← GeneratorTrasy(M, S, K)
                            route ← Trasa(M, temp)
                            comment Funkcja oblicza koszt trasy podróży
                            nowaTrasaKoszt ← TrasaKoszt(M, temp)

                            if nowaTrasaKoszt ‹trasa and route ≠ 1 then
                                    begin
                                            comment Zablokuj dostęp procesom do zmiennych
                                            koszt ← nowaTrasaKoszt
                                            trasa ← temp
                                            comment Odblokuj dostęp procesom do zmiennych
                                            comment Funkcja aktualizuje wartość feromonu
                                            RozłóżFeromon(feromon, trasa)
                                    end

                            comment Aktualizacja feromonu zmieniającego się w czasie
                            ParowanieFeromonu(feromon)
            endPARA
        write trasa
end
```



Rys. 1.2. Przypadek w którym mrówka nie może kontynuować podróży (mapa: http://osmapa.pl/)
Fig. 1.2. The case in which an ant cannot continue the journey (map: http://osmapa.pl/

Należy zauważyć, że może pojawić się wyjątek w działaniu algorytmu. Istnieje prawdopodobieństwo wystąpienia sytuacji, w której cyfrowa mrówka z obecnego węzła nie może udać się do kolejnego, gdyż już był wcześniej odwiedzony. W tej sytuacji, aby zapobiec zapętleniu algorytmu podróż mrówki zostanie przerwana, a trasa zostanie oznaczona jako błędna. Nie ma to żadnego wpływu na działanie pozostałych osobników w kolonii. Opisywany przypadek ilustruje rysunek 1.2.
Ogólną strukturę algorytmu opisywanego w tym artykule przedstawia listing 1.1. Prezentowany pseudokod został napisany w języku programowania wysokiego poziomu Pidgin ALGOL.

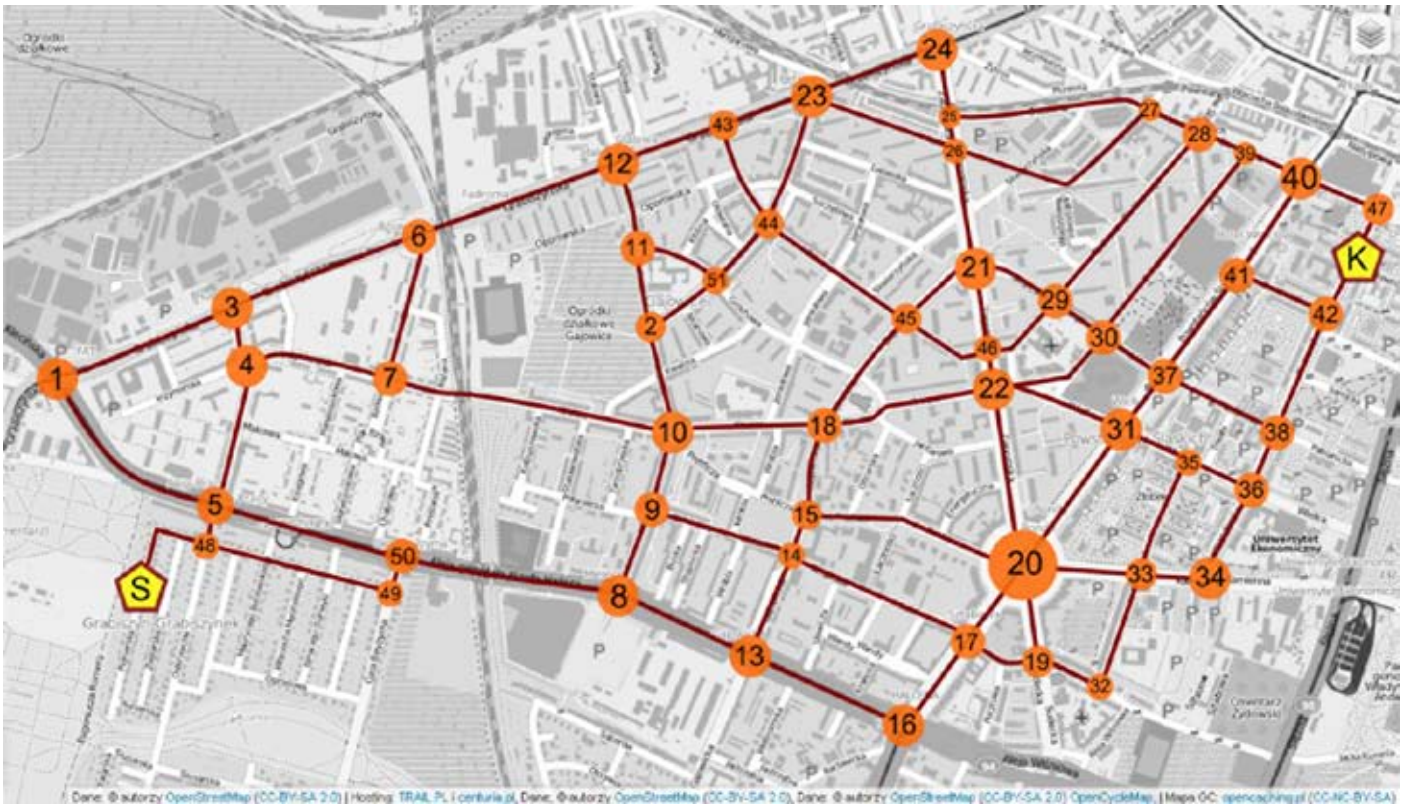
5. Eksperyment w warunkach rzeczywistych

Na potrzeby niniejszego artykułu z aglomeracji miejskiej wybrano fragment miasta Wrocław, w którym wytypowano istotne węzły komunikacyjne (rys. 1.3). Dzięki temu prostemu zbiegowi czytelnik uzyskał wyrazisty przekaz dotyczący metodologii przeprowadzanego badania.
Pomiędzy wszystkimi wyznaczonymi węzłami przeprwadzono analizę tras, umożliwiającą wyznaczenie wskaźników występowania korków i utrudnień drogowych. Eksperyment przeprowadzono wykorzystując dwa pojazdy, z czego jeden został wyposażony w tradycyjny system n wigacji, a drugi wykorzystywał system nawigacji[2] oparty na opisanym wcześniej algorytmie mrówkowym. Próbę wykonano w godzinach popołudniowego szczytu, gdyż w tym czasie poruszanie się po mieście sprawia najwięcej problemów ze względu na występujące utrudnienia w ruchu drogowym. Po wyznaczeniu trasy pojazdy wyruszyły jednocześnie z punktu oznaczonego symbolem „S" na fragmencie planu miasta i podążały do punktu docelowego oznaczonego symbolem „K".

Tab. 1.1. Wyznaczona trasa przejazdu przez tradycyjny system nawigacji
Tab. 1.1. The mapped travel route by the traditional navigation system

Trasa przejazdu	Szacowany czas
S, 48, 5, 50, 8, 13, 16, 17, 20, 31, 37, 41, 42, K	600 sekund

Tradycyjny system nawigacji wyznaczył statyczną trasę przez węzły (tab. 1.1) nie uwzględniając jakichkolwiek



Rys. 1.3. Fragment planu miasta z naniesionymi węzłami komunikacyjnymi (mapa: http://osmapa.pl/)
Fig. 1.3. Part of the city map with plotted traffic junctions (map: http://osmapa.pl/)

2 Autor tekstu używając sformułowania „tradycyjny system nawigacji" ma na myśli rozwiązanie nawigacyjne niewykorzystujące potencjału sztucznej inteligencji i informacji o panujących warunkach drogowych w czasie rzeczywistym, generujące statyczną trasę przejazdu.

warunków drogowych. Szacowany czas przejazdu całej trasy został określony na 600 sekund. Jednakże, rzeczywisty czas przejazdu wyniósł 1020 sekund, co wydłużyło przejazd o 70%. Na trasie przejazdu pojazd natrafił na utrudnienia drogowe pomiędzy węzłami: 50 – 8, 8 – 13, 13 – 16, 20 – 31, 31 – 37. Napotkane utrudnienia miały znaczący wpływ na rzeczywisty czas przejazdu, który znacznie odbiegał od szacowanego czasu. Pokonana trasa przez pojazd miała długość 4,3 km.

Tab. 1.2. Wyznaczone kolejne trasy przejazdu przez nowy system nawigacji

Tab. 1.2. The next mapped travel routes by the new navigation system

Trasa przejazdu	Szacowany czas
S, 48, 4, 7, 10, 18, 22, 31, 37, 41, 42, K	729.72 sekund
48, 5, 4, 7, 10, 9, 14, 15, 20, 33, 34, 36, 38, 42, K	664.64 sekund
5, 4, 7, 10, 18, 22, 31, 37, 41, 42, K	626.72 sekund
4, 7, 10, 18, 22, 31, 37, 38, 42, K	557.16 sekund
7, 10, 18, 22, 31, 37, 41, 42, K	503.52 sekund
10, 18, 22, 31, 35, 36, 38, 42, K	419.40 sekund
18, 22, 31, 35, 36, 38, 42, K	337.40 sekund
22, 31, 35, 36, 38, 42, K	255.00 sekund
31, 35, 36, 38, 42, K	179.08 sekund
35, 36, 38, 42, K	130.04 sekund
36, 38, 42, K	96.00 sekund
38, 42, K	63.76 sekund
42, K	30.08 sekund

Nowy system nawigacji uaktualniał trasę przejazdu w czasie rzeczywistym uwzględniając wszystkie czynniki utrudniające jazdę. Wyznaczone kolejne trasy przejazdu prezentuje tabela 1.2. Wyznaczanie nowej trasy przejazdu następowało pomiędzy węzłami wraz z nowo oszacowanym czasem przejazdu. Rzeczywisty czas przebycia drogi z punktu początkowego do docelowego wyniósł 780 sekund, co wydłużyło przejazd zaledwie o 6,89%, w stosunku do początkowo prognozowanej wartości. Trasa pokonana przez pojazd miała długość 4,2 km.

Pomiar również powtórzono w warunkach porannego szczytu oraz normalnego natężenia ruchu pomiędzy dwoma punktami „S" i „K" uzyskując następujący średni wynik. Dla wykorzystywanego obecnie systemu nawigacji średni realny czas przejazdu trwał 870 sekund, okres podróży zaś wydłużył się średnio o 47% w stosunku do szacowanego czasu. Wszak stosując nowe rozwiązanie uzyskano średnią wartość realnego czasu przejazdu równą 750 sekund i względem szacowanej wartości czasu, przejazd wydłużył się zaledwie o 2,78%.

Przeprowadzone eksperymenty uwzględniające topologię miasta i jego okolic na trasach przejazdu o długości powyżej 30 km wykazały, że dzięki nowej implementacji czas przejazdu uległ skróceniu w stosunku do ogólnodostępnego systemu nawigacji. Faktem jest, że każda z nawigacji wyznaczała indywidualną trasę przejazdu.

Dzięki temu, że były uwzględniane przez algorytm rzeczywiste warunki drogowe panujące w danym czasie z powodzeniem udało się ominąć występujące utrudnienia. Zmniejszeniu uległa również liczba negatywnych bodźców oddziałujących na stan psychofizyczny osoby kierującej pojazdem. Analizując wyniki badań przeprowadzonych eksperymentów można stwierdzić wyższość nowej implementacji algorytmu mrówkowego nad tradycyjnym rozwiązaniem.

Analizując wyniki badań przeprowadzonych eksperymentów można stwierdzić, że nowa implementacja algorytmu doskonale nadaje się do celów nawigacji w aglomeracjach miejskich.

6. Podsumowanie

Przeprowadzone eksperymenty mające na celu sprawdzenie funkcjonowania algorytmu w warunkach rzeczywistych zakończyły się pełnym sukcesem. Opracowany na potrzeby projektu system nawigacji wyznaczył trasę o krótszym czasie przejazdu. Tym samym przyczynił się na poprawę komfortu jazdy i wpłynął znacząco na samopoczucie kierowcy, co przekłada się również na większe bezpieczeństwo jazdy. Ponadto, z punktu widzenia ekonomicznego czas przejazdu wpływa również na ilość zużytego paliwa. Natomiast krótszy czas przejazdu to nie tylko oszczędności, ale również zmniejszona emisja zanieczyszczeń do atmosfery.

Poszukiwanie najlepszej trasy przejazdu będzie w przyszłości najważniejszym elementem nowoczesnych systemów nawigacji. Algorytmy poszukujące optymalnego rozwiązania z uwzględnieniem wielu parametrów mogą wpłynąć znacznie na zmniejszenie zatorów na drogach w dużych aglomeracjach miejskich poprzez kierowanie ruchu alternatywnymi trasami. Również urządzenia nawigacyjne posiadające wbudowane procesory wielordzeniowe pozwolą na stosowanie rozwiązań przetwarzania równoległego, co znacznie skróci czas generowania rozwiązań w czasie rzeczywistym. Z biegiem czasu stanie się również możliwe pobieranie danych o przepustowości ruchu drogowego dzięki montażowi odpowiednich urządzeń pomiarowych na skrzyżowaniach.

Możliwe, że w bliskiej przyszłości do wyznaczania trasy pojazdów będzie się używać algorytmów niedeterministycznych, które umożliwią wyznaczenie optymalnego w danych warunkach rozwiązania.

Warto dodatkowo zauważyć, że w obecnych czasach do rozwiązania przedstawionego problemu niewystarczająca okazuje się wszechstronna wiedza z jednej, konkretnej

dziedziny nauk. Powstaje więc wrażenie, że zagadnienie jest w swojej istocie bardzo skomplikowane i rzeczywiście w praktyce wymaga zastosowania wiedzy z dziedzin nauk behawioralnych, ścisłych, i współczesnej techniki. Powoduje to, że osoba pracująca nad takim zagadnieniem musi się charakteryzować nieprzeciętną erudycją lub wymagane jest powstanie zespołów badawczych.

Literatura (References)

[1] E. Bonabeau, M. Dorigo, G. Theraulaz, *Swarm Intelligence From Natural to Artificial Systems*, Oxford University Press 1999.
[2] J. Bąk, *Psychologiczne badania kierowców.* Bezpieczeństwo pracy nr 6, pp.12-15, 2004.
[3] M. Dorigo, L. Maria Gambardella, *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*, IEEE Transactions on Evolutionary Computation, Vol.1, No.1, pp.53-66, 1997.
[4] M. Dorigo, T Stützle, *Ant Colony Optimization*, MIT Press, Cambridge 2004.
[5] M. T. Jones, *Artificial Intelligence: A Systems Approach*, Infinity Science Press LLC 2008.
[6] W. Mizerski, W. Sadowski, A. Garbarczyk, B. Tokarska, K. Mazur, *Tablice matematyczne*, Adamantan, Warszawa 2008.

Hybrydowy algorytm mrówkowy wykorzystujący algorytm genetyczny do wyznaczania trasy w systemie nawigacji

A hybrid ant algorithm using genetic algorithm to determine the route in navigation system

Daniel Komar[1]

Treść. Artykuł ma na celu zaprezentowanie nowej implementacji hybrydowego algorytmu mrówkowego, który do rozwiązywania postawionego problemu wyznaczenia optymalnej trasy przejazdu będzie wykorzystywał również algorytm genetyczny. Autor przedstawi wyniki symulacji przeprowadzonej na podstawie rzeczywistych danych, ukazując znaczny wzrost efektywności rozwiązywania problemu. Otrzymane wyniki wykazały, że nowy algorytm wyznaczał w większej liczbie przypadków znacznie krótszy czas przejazdu, a tym samym redukował występujące czynniki zakłócające i negatywnie wpływające na osobę kierującą pojazdem.

Słowa kluczowe: algorytm mrówkowy, algorytm genetyczny, system nawigacji

Abstract. The purpose of this paper is to present the new implementation of a hybrid ant algorithm that will also use a genetic algorithm in order to solve the problem consisting in optimal route calculation. The author will present results of simulations that were performed based on real data and showed a significant increase of problem solution effectiveness. The obtained results proved that the new algorithm determined in more number of cases a significantly shorter journey time and in consequence reduced the occurring confounding factors which had a negative impact on the person driving a vehicle.

Keywords: ant algorithm, genetic algorithm, navigation system

1. Wstęp

Algorytm mrówkowy i genetyczny jest stosunkowo nową techniką optymalizacji pozwalającą rozwiązać problem poszukiwania optymalnej trasy przejazdu. Nowoczesne implementacje tych algorytmów pozwalają rozwiązywać problem, określony w teorii złożoności obliczeniowej jako NP-trudny. Do tej klasy problemu możemy zaliczyć wyznaczanie trasy przejazdu pomiędzy dwoma punktami i problem chińskiego listonosza (*ang. Chinese postman problem*).

Podstawowym założeniem algorytmu mrówkowego jest naśladowanie zachowania kolonii mrówek występujących w realnym świecie. Natomiast ich odpowiednikiem w cyfrowej rzeczywistości są wygenerowane mrówki, które będą w ograniczonym zakresie dokonywały oceny alternatywnych wariantów w procesie decyzyjnym [2, 3]. Algorytm genetyczny z założenia naśladuje ewolucję osobników, występujących w przyrodzie. Podczas działania wykonuje podstawowe operacje krzyżowania i mutacji [4, 5]. Następnie na podstawie cyfrowego kodu genetycznego chromosomów, obliczana jest funkcja przystosowania każdego z osobników. Oba algorytmy bazują na podstawowych mechanizmach działania zachodzących w świecie przyrody, na podstawie których sformułowano ich główne zasady. Prekursorom poszukiwania nowych rozwiązań technicznych obserwacje świata natury dały wiele pomysłów mających we współczesnych czasach praktyczne zastosowanie w dziedzinie sztucznej inteligencji, jak również algorytmice. Twórcą algorytmu genetycznego jest John Henry Holland.

W artykule zostanie przedstawiony hybrydowy algorytm mrówkowy, który do wyznaczania optymalnej trasy przejazdu pomiędzy punktem początkowym i docelowym, będzie wykorzystywał algorytm genetyczny. Powstała nowa implementacja opisywanego rozwiązania będzie uwzględniała również wiele istotnych czynników zmieniających się w czasie rzeczywistym.

W dalszej części zostanie zaprezentowana symulacja działania nowego algorytmu przeprowadzona na rzeczywistych danych. Ma ona na celu sprawdzenie możliwości zwiększenia efektywności generowania optymalnego rozwiązania zadanego problemu.

2. Tło koncepcji

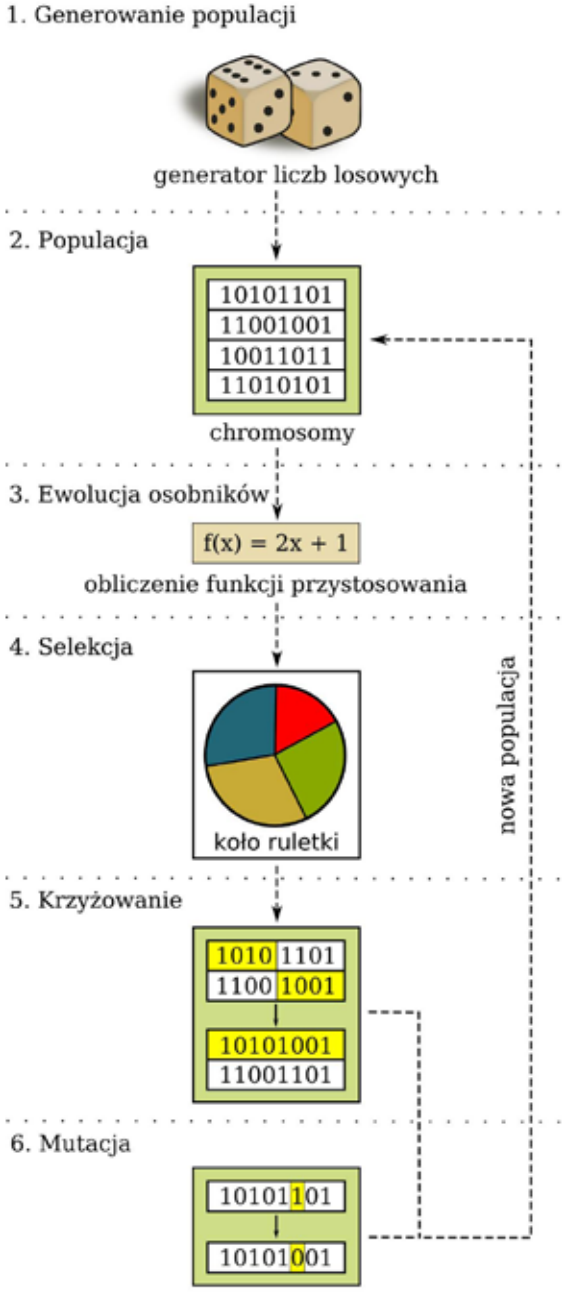
Weną twórczą do rozpoczęcia prac nad nową hybrydową implementacją algorytmu były prace psychologiczne Jadwigi Bąk prowadzone w Zakładzie Psychologii Transportu Drogowego Instytutu Transportu Samochodowego w Warszawie. Prowadzone badania dotyczyły wpływają-

1 Wydział Informatyki, Wrocławska Wyższa Szkoła Informatyki Stosowanej, ul. Wejherowska 28, 54-239 Wrocław, gordon1x@poczta.fm

cych na osobę kierującą pojazdem negatywnych czynników występujących w ruchu drogowym [1].
Głównym celem nowej implementacji jest dobieranie trasy przejazdu w taki sposób, aby zminimalizować liczbę negatywnych czynników występujących w otoczeniu podczas podróży. Z założenia miałoby to pozytywny wpływ na indywidualny stan emocjonalny kierowcy, poprawę bezpieczeństwa i zachowania wobec innych uczestników ruchu. Istotne w tym przedsięwzięciu jest ograniczenie liczby iteracji algorytmu, dostarczającego optymalne w danych warunkach rozwiązanie, gdyż czas potrzebny na wygenerowanie nowej trasy jest ściśle określony. Wynika to z faktu, że kierowca widzący nową informację musi mieć czas na podjęcie odpowiedniej decyzji i reakcji przed dojazdem do węzła komunikacyjnego.

3. Algorytm genetyczny

W prezentowanym rozwiązaniu istotne znaczenie odgrywa algorytm genetyczny, który został zaadaptowany do działania z algorytmem mrówkowym. Jest to algorytm ewolucyjny, będący stochastyczną metodą poszukiwania rozwiązania, bazującego na procesie naturalnej selekcji i ewolucji osobników zachodzącym w realnym środowisku [4, 5]. Każdy osobnik w populacji jest reprezentowany przez chromosom składający się ze zbioru symboli. Głównym założeniem algorytmu jest imitowanie czynności związanych z procesem manipulacji genami, do których można zaliczyć wykonywanie operacji krzyżowania i mutacji. Realizowanie elementarnych manipulacji na wygenerowanych chromosomach pozwoli tworzyć kolejne generacje osobników. Natomiast proces selekcji zostanie uzyskany poprzez odpowiedni dobór osobników do operacji krzyżowania.
W zaimplementowanym na potrzeby projektu algorytmie zbiór symboli składa się ze zbioru liczb naturalnych z zerem włącznie. Pierwszym i kluczowym etapem jest wygenerowanie początkowej populacji, który następuje tylko raz, podczas inicjacji działania algorytmu. Wygenerowane kolejno losowe wartości przez generator liczb losowych są wprowadzane do macierzy, reprezentującej chromosomy poszczególnych osobników tworząc populację. W kolejnej fazie jest obliczana wartość funkcji przystosowania. Wspomniana funkcja służy do znalezienia najlepszego rozwiązania oraz jej wynik wykorzystywany jest do wyznaczania indywidualnego prawdopodobieństwa krzyżowania poszczególnych par osobników. Wyższa wartość funkcji przystosowania danego osobnika zwiększa jego prawdopodobieństwo krzyżowania z innymi osobnikami w populacji. Następnie zostają określone procentowe przedziały dla każdego osobnika na tzw. kole ruletki i zostają wybrane pary osobników do krzyżowania przy pomocy generatora liczb losowych. Dzięki tej czynności imitowany jest proces naturalnej selekcji występujący w przyrodzie. W kolejnym kroku rozpoczyna się proces krzyżowania i mutacji. Proces krzyżowania dwóch osobników jest indywidualną rekombinacją ich cyfrowego kodu genetycznego, który tworzy nowy chromosom osobnika kolejnej generacji. Proces mutacji może zachodzić podczas krzyżowania. Dzięki tej elementarnej operacji zostaje wprowadzona zmienność w kodzie chromosomów nowych osobników. W efekcie funkcja przystosowania osobników nie tylko w kolejnych generacjach ulega zwiększeniu, ale również może skutkować jej zmniejszeniem, co może prowadzić tym samym do znalezienia nowego lepszego niż poprzednio dostarczonego rozwiązania przez algorytm. Zwieńczeniem wykonania wszystkich opisanych operacji jest powstanie nowej populacji osobników.
Opisane kroki wykonywanych czynności zostały zaprezentowane w uproszczeniu na schemacie (rys. 1.1). Z kolei ogólna struktura algorytmu genetycznego została zaprezentowana na listingu 1.1. Uproszczony pseudokod algorytmu genetycznego został napisany w języku programowania wysokiego poziomu Pidgin ALGOL.



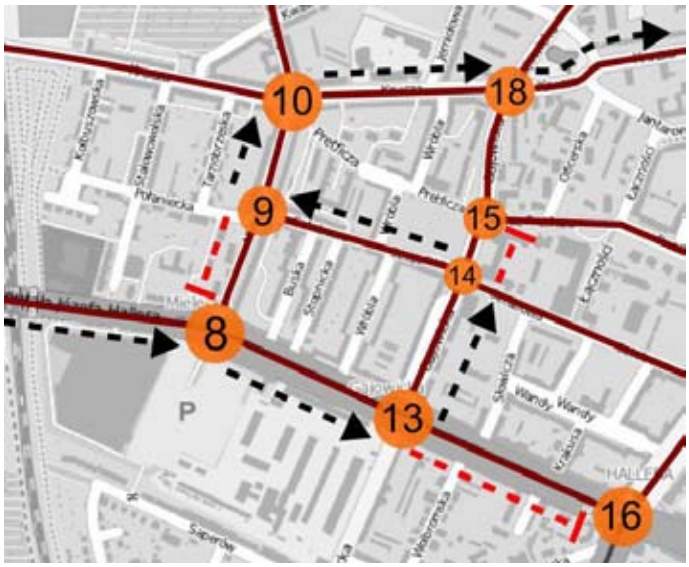
Rys. 1.1. Uproszczony schemat funkcjonowania algorytmu genetycznego
Fig. 1.1. A simplified scheme of the genetic algorithm working principle

List. 1.1. Uproszczony pseudokod algorytmu w języku programowania Pidgin ALGOL
List. 1.1. The simplified pseudocode of the algorithm in Pidgin ALGOL programming language

```
procedure AlgorytmGenetyczny(G, C):
     begin
          comment G – liczba osobników, C – macierz osobników,
          comment X – wielkość chromosomu osobnika
          read X
          comment Zmienne wymagane przez algorytm genetyczny
          operator ← Ø
          węzły    ← Ø

          comment Funkcja wyznaczająca rozwiązanie zadanego problemu
          węzły ← Rozwiązanie(C)

          comment Funkcja obliczająca funkcję przystosowania dla każdego osobnika
          operator ← Przystosowanie(G, operator)

          comment Funkcja obliczająca procentowe przedziały dla każdego osobnika
          operator ← KołoRuletki(G, operator)

          comment Funkcja losująca osobniki do krzyżowania
          pary ← Pary(G, operator)

          comment Funkcja dokonująca krzyżowania i mutacji, która
          comment w wyniku zwraca następną generację osobników
          C ← KrzyżowanieMutacja(G, pary)

          return węzły
     end
```

4. Hybryda algorytmu mrówkowego

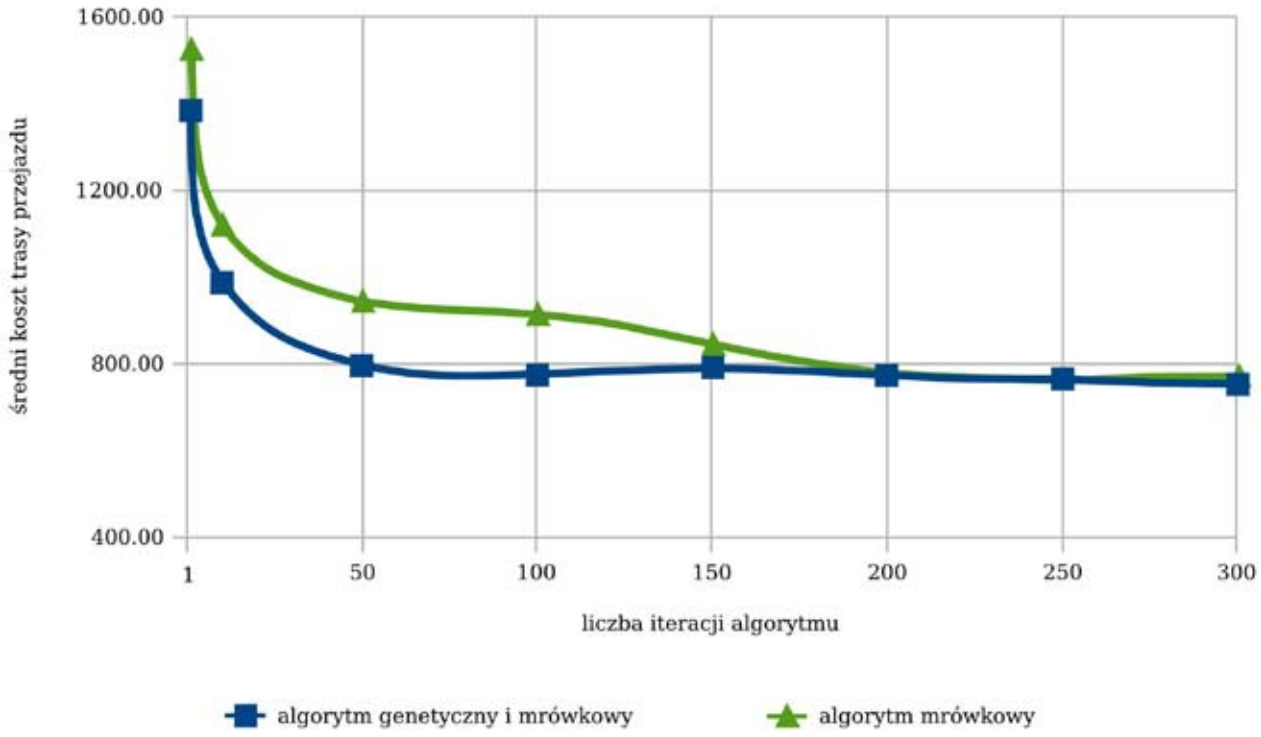
W opracowanej implementacji hybrydowej algorytmu mrówkowego główną rolę będzie odgrywał już wcześniej wspomniany algorytm genetyczny. Należy również w tym momencie wyjaśnić zasadę działania modułu algorytmu genetycznego, który ułatwi cyfrowej mrówce wybór węzłów o niższym koszcie. W tradycyjnych rozwiązaniach algorytm genetyczny wykorzystuje się do poszukiwania najlepszego rozwiązania. Niestety w tym przypadku będzie on działał zupełnie inaczej, ponieważ nie będzie on poszukiwał najniższego kosztu podróży pomiędzy węzłami na mapie, lecz będzie wykorzystywany do wyznaczenia tych najdroższych odcinków komunikacyjnych. Uzyskany przez najlepszego osobnika wynik zostanie wykorzystany do blokowania możliwości wyboru takiego połączenia przez cyfrową mrówkę (rys. 1.2) i tym samym niezależnie od wartości rozłożonego feromonu na trasie będzie ona zmuszona wybrać lokalnie występujące inne lepsze rozwiązanie. Z założenia powinno to znacznie poprawić skuteczność generowania lepszych rozwiązań.
Powstała implementacja połączenia obu algorytmów może z powodzeniem zostać wykorzystana w systemach wieloprocesorowych, przetwarzania równoległego i rozproszonego. Dzięki zastosowaniu nowych technologii przetwarzania danych znacznej redukcji ulega czas generowania rozwiązań.



Rys. 1.2. Blokowanie możliwości wyboru trasy pomiędzy węzłami (mapa: http://osmapa.pl)
Fig. 1.2. Blocking of route selection between traffic junctions (map: http://osmapa.pl/)

5. Symulacja działania nowego rozwiązania

Do wykonania symulacji posłużono się rzeczywistymi danymi, które zostały zebrane na podstawie analizy ruchu na danych węzłach komunikacyjnych. Wszystkie główne



Wyk. 1.1. Wykres prezentujący porównanie efektywność dwóch algorytmów
Fig. 1.1. The chart showing comparison of the efficiency of two algorithms

węzły zostały zaznaczone na fragmencie miasta Wrocław. Zarówno nowy hybrydowy algorytm, jak i algorytm mrówkowy miał wygenerować trasę przejazdu przy ściśle określonej liczbie iteracji. Dla zadanej liczby iteracji wykonano kilkanaście prób i obliczono ich średnią wartość. Efektywność wyznaczania optymalnej trasy przez dany algorytm prezentuje wykres 1.1.

Połączenie metodologii dwóch różnych algorytmów względem poprzednio stosowanego rozwiązania, skutkuje znacznym zmniejszeniem liczby wymaganych iteracji do wyznaczenia optymalnego wyniku. Można również stwierdzić, że w większości przypadków hybrydowy algorytm generował trasę o znacznie niższym koszcie, a zwiększając liczbę iteracji jego wzrost efektywności [w pewnym okresie] utrzymywał się w przybliżeniu na stałym poziomie. Należy również mieć na uwadze, że otrzymane wyniki zależą od liczby osobników w populacji. W symulacji populacja liczyła dwudziestu ośmiu osobników i została ona uznana za najlepszą na podstawie wcześniej wykonanych testów.

Możemy również zobaczyć jak zmieniała się funkcja przystosowania w kolejnych generacjach osobników na wykresie 1.2. Z wykresu wynika, że dążyła tylko do maksymalizacji wartości najlepszego osobnika, ale również ulegała znacznym wahaniom wartości. Wspomniane oscylacje funkcji przystosowania są spowodowane występującymi mutacjami podczas krzyżowania chromosomów danych osobników.

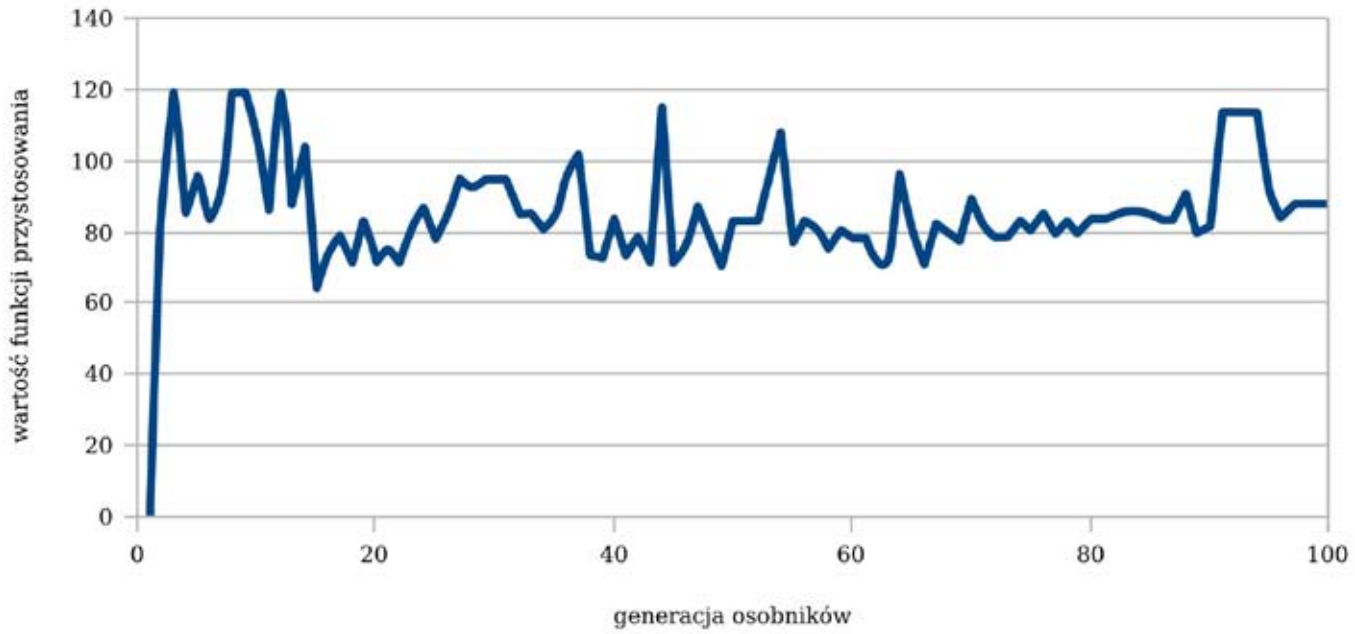
Z obu wykresów możemy błędnie wywnioskować, że wahania wartości funkcji przystosowania mają znikomy wpływ na wyznaczony koszt trasy przez cyfrową mrówkę. Wbrew pozorom zmienność zachodząca w populacji poprzez efekt mutacji jest zamierzona i ma za zadanie wyznaczyć jedne z droższych tras pomiędzy różnymi węzłami komunikacyjnymi. Następstwem tego jest nieutrzymywanie stałej osiągniętej wartości jako najlepszej, gdyż istnieje prawdopodobieństwo, że cyfrowa mrówka pomiędzy najdroższymi węzłami może się nie poruszać. Wyznaczenie o dużej wartości wag lokalnych połączeń pomiędzy dowolnymi węzłami zostaje osiągnięte poprzez operacje mutacji, co znacząco wpływa na proces decyzyjny i wynik generowany przez algorytm mrówkowy.

Na uwagę zasługuje fakt, że prezentowane w artykule wyniki zostały obliczone na podstawie wykonanych prób. Natomiast podczas wykonywania kolejnych symulacji wyniki mogą od siebie odbiegać. Wynika do z faktu, że są to algorytmy niedeterministyczne, a otrzymany wynik obliczeń jest indywidualny i może się każdorazowo różnić od poprzedniego. Symulację powtórzono kilkakrotnie i za każdym razem średnie wartości generowane przez algorytm, który wykorzystywał połączenie dwóch metod, uzyskiwał niższą wartość.

6. Podsumowanie

Zaprezentowane w artykule wyniki przeprowadzonych symulacji dowiodły, że połączenie możliwości obu algorytmów daje zaskakujące rezultaty. Powstała nowa implementacja pozwoliła na wyznaczenie optymalnej trasy przejazdu ze znacznie większą efektywnością przy stosunkowo małej liczbie iteracji. Wykorzystanie takich rozwiązań daje znaczne możliwości, ponieważ pozwala modyfikować na bieżąco trasę przejazdu pomiędzy węzłami w znacznie krótszym czasie.

Zastosowanie prezentowanego rozwiązania w systemie



Wyk. 1.2. Wykres prezentujący oscylację funkcji przystosowania w kolejnych generacjach osobników
Fig. 1.2. The chart showing the oscillation of fitness function in the consecutive generations of new individuals

nawigacji pozwala bezpośrednio wpływać na stan emocjonalny i psychofizyczny kierowcy. Poprzez ograniczenie występujących w otoczeniu negatywnie wpływających czynników zakłócających zwiększa się bezpieczeństwo i komfort podróży. Ponadto w odbieranych bodźcach komunikacyjnych istotne stają się sygnały z otoczenia, które wymuszają na kierowcy podejmowanie istotnych decyzji. Zasadniczą rolę odgrywa proces antycypacji zdarzeń, umiejętnego wykorzystania schematów działania czy w wielu przypadkach myślenia abstrakcyjnego.

Możliwe, że podobne połączenia algorytmów w bliskiej przyszłości będą wykorzystywane w autonomicznych pojazdach do wyznaczania optymalnej trasy przejazdu, która może być modyfikowana w czasie rzeczywistym. Również tego typu algorytm pozwala na zwiększenie efektywności procesu decyzyjnego w określonym lokalnym środowisku, co może się przekładać na zachowanie takiego pojazdu na pewnym odcinku, wśród obecności innych uczestników ruchu. Musimy pamiętać, że możliwości takich algorytmów nie ograniczają się tylko do problemu wyznaczania trasy, a ich zastosowanie w dużej mierze zależy od wizji projektanta danego rozwiązania.

Literatura (References)

[1] J. Bąk, *Psychologiczne badania kierowców.* Bezpieczeństwo pracy nr 6, pp.12-15, 2004.

[2] M. Dorigo, L. M. Gambardella, *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*, IEEE Transactions on Evolutionary Computation, Vol.1, No.1, pp.53-66, 1997.

[3] M. Dorigo, T. Stützle, *Ant Colony Optimization*, MIT Press, Cambridge 2004.

[4] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge 1999.

[5] R. L. Haupt, S. E. Haupt, *Practical Genetic Algorithms,* 2nd Edition, John Wiley & Sons 2004.

Weryfikacja „słabej” hipotezy Goldbacha do 10^{31}

Verifying the „weak” Goldbach conjecture up to 10^{31}

Łukasz Świerczewski[1]

Treść. Praca prezentuje aspekt numerycznej weryfikacji „słabej” hipotezy Goldbacha dla wartości mniejszych niż 10^{31}. Do obliczeń, które zajęły w sumie ok. 50 000 godzin czasu pojedynczego CPU wykorzystano klaster wydajnościowy złożony z procesorów AMD Opteron 4284. Podczas sprawdzania pierwszości zastosowano test Millera-Rabina. Przetestowano także możliwe zastosowanie testu ECPP. Jak się okazało przy założeniu dodatkowych warunków poprawności testu Millera-Rabina „słaba” hipoteza Goldbacha w badanym zakresie jest prawidłowa.

Słowa kluczowe: teoria liczb, hipoteza Goldbacha, liczby pierwsze

Abstrackt. This paper presents aspect of the numerical verification a „weak” Goldbach's conjecture for values less than 10^{31}. For calculations, that took about 50 000 hours of a single CPU performance, there was used an performance cluster consisting of the AMD Opteron 4284 processors. During the primality check, there was used Miller-Rabin test. There was also tested the possiblity of ECPP test usage. As it turned out, when there were added some additional conditions of correctness of Miller-Rabin test, the „weak” Goldbach's conjecture occurs correct in researched range.

Key words: number theory, Goldbach conjecture, primes

1. Wprowadzenie

Hipoteza Goldbacha zakłada, że każdą liczbę parzystą większą od 2 można przedstawić jako sumę dwóch liczb pierwszych. Problem ten pierwotnie sformułował Goldbach w liście do Eulera w 1742 roku. Istnieje także tzw. „słaba” hipoteza Goldbacha mówiąca że każda liczba naturalna nieparzysta większa od 7 jest sumą trzech nieparzystych liczb pierwszych *(niekoniecznie różnych).* W 1937 roku Iwan Vinogradov udowodnił [1], że każdą dostatecznie dużą liczbę nieparzystą można przedstawić w postaci sumy trzech liczb pierwszych. Wynik ten poprawili w 2002 roku Liu Ming-Chit i Wang Tian-Ze z Uniwersytetu w Hong Kongu [2]. Udowodnili oni, że „dostatecznie duża” liczba oznacza większa od e^{3100} (w przybliżeniu 10^{1346}).

2. Algorytm

Wykorzystany algorytm został szczegółowo opisany w publikacji Yannick'a Saouter'a z 1998 roku [3]. Wygląda on następująco:

1. Do p_0 wpisz 100000000209366024193 i przejdź do kroku 2.
2. Jeżeli p_i jest liczbą pierwszą to inkrementuj i oraz p_i zwiększ o $95367431640 \cdot 2^{22}$ i przejdź do kroku 4. W przeciwnym wypadku przejdź do kroku 3.
3. Inkrementuj i oraz p_i zmniejsz o $10 \cdot 2^{22}$ i przejdź do kroku 2.
4. Jeżeli $p_i < 10^{31}$ wykonuj dalej obliczenia i przejdź do kroku 2. W przeciwnym wypadku przerwij działanie programu.

W pierwotnym artykule w kroku pierwszym algorytmu do p_0 została przypisywana wartość 138412033, a w kroku drugim p_i było zwiększane o $95360 \cdot 2^{22}$. Dane te jednak zaktualizowano biorąc pod uwagę najnowsze osiągnięcia. Wartość 138412033 została zwiększona ze względu na rozpoczęcie obliczeń tam gdzie zostały one w publikacji [3] zakończone. Zastąpienie $95360 \cdot 2^{22}$ wartością $95367431640 \cdot 2^{22}$ wynika z tego, że na czas pisania artykułu „mocną” hipotezę Goldbacha dwukrotnie sprawdzono już do $4 \cdot 10^{17}$ [4] (jednokrotnie aż do $4 \cdot 10^{18}$). $95367431640 \cdot 2^{22}$ jest największą wielokrotnością liczby $10 \cdot 2^{22}$ mniejszą od $4 \cdot 10^{17}$.

3. Implementacja algorytmu

Do obliczeń wykorzystano algorytm zaimplementowany z wykorzystaniem zoptymalizowanej pod kątem wykorzystanych procesorów biblioteki GMP (GNU Multiple Precision Arithmetic Library) [5]. Podczas testowania pierwszości zastosowano test Millera-Rabina [6] z określonymi świadkami pierwszości - wybrano w tym celu 20 najmniejszych liczb pierwszych. Tego typu rozwiązanie zostało teoretycznie rozpatrzone przez Zhang'a [7] i w połączeniu z testem Millera-Rabina można w ten sposób uzyskać bardzo szybki test pierwszości dla liczb mniejszych niż 10^{36}.

Przetestowano także wydajność rozwiązania wykorzystującego dodatkowo test ECPP [9]. W tym algorytmie początkowo sprawdzano pierwszość z wykorzystaniem testu Millera-Rabina, a jeżeli okazało się, że dana liczba jest

1 Instytut Informatyki i Automatyki, Państwowa Wyższa Szkoła Informatyki i Przedsiębiorczości w Łomży, lswierczewski@pwsip.edu.pl

według niego pierwsza wykonywano dodatkowo całkowicie deterministyczny test ECPP. Program wykorzystujący także ECPP okazał się znacząco wolniejszy i z jego wykorzystaniem zweryfikowano hipotezę tylko do 10^{24}. Na Listingu 1 przedstawiono główną część kodu realizującego weryfikację „słabej" hipotezy Goldbacha. Kod ten został napisany w języku C i standardzie OpenMP umożliwiającym wykonywanie obliczeń na komputerach równoległych z pamięcią wspólną. Za pomocą makr MILLER_RABIN oraz ECPP w przypadku tego listingu zdefiniowane jest użycie różnych algorytmów testowania pierwszości. Ze względu na ograniczone miejsce w tej pracy nie zaprezentuję kodu odpowiedzialnego za przetwarzanie danych na klastrach komputerowych. Jest on analogiczny do tego działającego z wykorzystaniem OpenMP lecz zostały w nim użyte funkcje interfejsu MPI.

```
    #pragma omp parallel shared(S, add_1, add_2, number_of_threads) private(thread_id, number, upper_range, counter)
   {
      thread_id = omp_get_thread_num();

      mpz_set(number, table_down[thread_id]);
      mpz_set(upper_range, table_upper[thread_id]);

      while( mpz_cmp(number, upper_range) == -1 )
      {

   # ifdef MILLER_RABIN
         if(miller_rabin(number, S) == 1)
   # endif
   # ifdef ECPP
         if(gmp_ecpp(mpz_class(number)) == 1)
   # endif
         {
            mpz_add_ui(number, number, add_1);
         }
         else
         {
            mpz_sub_ui(number, number, add_2);
         }
      }
   }
```

Listing 1. Implementacja części głównej kodu realizującego weryfikację „słabej" hipotezy Goldbacha

Listing 1. The implementation of the main part of the code performing a „weak" Goldbach's Conjecture verification.

Na Listing 2 została zaprezentowana funkcja główna realizująca test Millera-Rabina. Jako argumenty przyjmuje ona liczbę, która jest testowana oraz tablicę liczb pierwszych. Funkcja ta wywołuje funkcję pomocniczą miller_rabin_pass(m, k), która wykonuje test liczby m względem świadka pierwszości k.

```
int miller_rabin(mpz_t n, char *S)
{

   mpz_t pom;
   mpz_init(pom);
   mpz_set_ui(pom, 2);

   if(miller_rabin_pass(pom, n) == 0)
   {
      mpz_clear(pom);
      return 0;
   }


   for(unsigned short int i = 3; i < 72; i += 2)
   {
      if(S[i] == 1)
      {
         mpz_set_ui(pom, i);

         if (miller_rabin_pass(pom, n) == 0)
         {
            mpz_clear(pom);
            return 0;
         }
      }
   }
   mpz_clear(pom);

   return 1;
}
```

Listing 2. Implementacja funkcji głównej realizującej test pierwszości Millera-Rabina

Listing 2. The implementation of the main function performing Miller-Rabin primality test.

4. Obliczenia

Obliczenia wykonano na klastrze wydajnościowym złożonym z procesorów AMD Opteron 4284 i zajęły one w sumie 50000 godzin czasu pracy pojedynczego CPU. W oprogramowaniu wykorzystano środowisko MPI [12]. Ostatnią liczbą pierwszą jaką wygenerował program używający testu Millera-Rabina była 100000000000003998 61783893901313. W Tabeli 1 przedstawiono 30 ostatnich liczb pierwszych jakie wygenerowało oprogramowanie wykorzystujące test Millera-Rabina. Analogiczną tabelą dla rozwiązania całkowicie deterministycznego opartego także na teście ECPP jest Tabela 2.

Tab. 1. Ostatnich 30 liczb jakie wygenerował algorytm wykorzystujący test Millera-Rabina.
Tab. 1. The last 30 numbers generated by the algorithm which uses Miller-Rabin test.

999999999998879986181739852595 3	999999999999479986179764597555 3
999999999998919986181676675891 3	999999999999519986179697226547 3
999999999998959986181655442227 3	999999999999559986179487249203 3
999999999998999986181290275635 3	999999999999599986179319214899 3
999999999999039986181147407155 3	999999999999639986179247649587 3
999999999999079986181126173491 3	999999999999679986179234804531 3
999999999999119986180949750579 3	999999999999719986178991272755 3
999999999999159986180932711219 3	999999999999759986178970039091 3
999999999999199986180731122483 3	999999999999799986178969776947 3
999999999999239986180546310963 3	999999999999839986178818519859 3
999999999999279986180546048819 3	999999999999879986178654679859 3
999999999999319986180407374643 3	999999999999919986178654417715 3
999999999999359986180352586547 3	999999999999959986178498966323 3
999999999999399986180176163635 3	999999999999999986178389652275 3
999999999999439986180092015411 3	100000000000003998617838939013 13

Tab. 2. Ostatnich 30 liczb jakie wygenerował algorytm wykorzystujący test Millera-Rabina oraz ECPP.
Tab. 2. The last 30 numbers generated by the algorithm which uses Miller-Rabin and ECPP tests.

999988699999803877097473	999994699999792429268993
999989099999802658127873	999950999997920072171 53
999989499999802110246913	999954999997916690513 93
999989899999802107625473	999958999997891498475 53
999990299999801853345793	999996299999789105283073
999990699999801599066113	999996699999788767117313
999991099999801009242113	999997099999787380375553
999991499999800754962433	999997499999786706665473
999991899999800249024513	999997899999785697411073
999992299999800246403073	999998299999785065644033
999992699999799950180353	999998699999784685535233
999993099999795669368833	999999099999783969882113
999993499999794408456193	999999499999783422001153
999993899999793399201793	999999899999781406113793
999994299999792515776513	100000029999978140349235 3

5. Wnioski

Osiągnięcie granicy 10^{1346} jest całkowicie nierealne przy zastosowaniu współczesnych algorytmów i technik przetwarzania danych. Udowodniono, że przy założeniu poprawności uogólnionej hipotezy Riemmana granica ta może zostać obniżona do $3.2 \cdot 10^{49}$ [8] co jednak także nie będzie w naszym zasięgu w ciągu najbliższych lat. Postęp w badaniach dotyczących hipotezy Goldbacha jest jednak ogromny. W sierpniu 2012 roku Agostino Prástaro opublikował dowód „mocnej” hipotezy Goldbacha [10], a w maju 2013 roku H. A. Helfgott w swoim artykule [11] zamieścił dowód „słabej” hipotezy Goldbacha dla wartości większych niż 10^{29}. Obydwa dowody jednak na czas pisania niniejszej publikacji nie przeszły pozytywnie weryfikacji. Niniejsza praca jest jest drobnym krokiem ku potwierdzeniu poprawności hipotezy Goldbacha. Zaimplementowane na potrzeby artykułu kody programów mogą w przyszłości umożliwić weryfikację hipotezy w jeszcze większym zakresie.

Literatura (References)

[1] I.M. Vinogradov, Representation of an odd number as the sum of three primes, Dokl. Akad. Nauk SSSR (1937), no. 15, 169-172.
[2] MC Liu, T. Z. Wang, On the Vinogradov bound in the three primes Goldbach conjecture, Acta Arithmetica 105 (2002): 133-175.
[3] Y. Saouter, Checking the odd Goldbach conjecture up to 10^{20}, Mathematics of Computation of the American Mathematical Society 67.222 (1998): 863-866.
[4] Tomás Oliveira e Silva, http://sweet.ua.pt/tos/goldbach.html
[5] T. Granlund. The GNU Multiple Precision Arithmetic Library. TMG Datakonsult, Boston, MA, USA, 2.0.2 edition, June 1996.
[6] J. Hurd, Verification of the Miller–Rabin probabilistic primality test, The Journal of Logic and Algebraic Programming 56.1 (2003): 3-21.
[7] Z. Zhang, Two Kinds of Strong Pseudoprimes up to 10^(36), Math. Comput. 76, 2095-2107, 2007.
[8] T.Z. Wang, J.R. Chen, On odd Goldbach problem under general Riemann hypothesis ,Sci. China Ser. A 36 (1993), no. 6, 682-691. MR 95a:11090.
[9] J. Franke, et al., Proving the Primality of Very Large Numbers with fastECPP, Algorithmic Number Theory. Springer Berlin Heidelberg, 2004. 194-207.
[10] A. Prástaro, The Goldbach's conjecture proved, *arXiv preprint arXiv:1208.2473* (2012).
[11] H. A. Helfgott, Major arcs for Goldbach's theorem, *arXiv preprint arXiv:1305.2897* (2013).
[12] W. D. Gropp, E. Lusk and A. Skjellum, *Using MPI: portable parallel programming with the message-passing interface*. Vol. 1. the MIT Press 1999.

Intel Manycore Testing Lab - środowisko sprzętowo-programowe do dydaktyki tworzenia i testowania efektywności równoleglizacji oprogramowania

Intel Manycore Testing Lab - hardware and software environment focused on didactic of development and efficiency testing in software paralleling

Łukasz Świerczewski[1]

Treść. Współczesny proces dydaktyczny technik programowania często wymaga dostępu zarówno do nowoczesnego sprzętu, jak i oprogramowania. W szczególnej mierze odnosi się to do algorytmów równoległych, których odpowiednie właściwości w dużo większym stopniu można zaobserwować na wydajnych procesorach nowej generacji. Aby stworzyć międzynarodową społeczność akademicką związaną z tą specjalizacją firma Intel udostępniła wirtualne laboratorium testowe (Manycore Testing Lab - MTL). Artykuł przedstawia aspekt architektury oraz praktycznego zastosowania MTL w pracy wielożytkowej i skupia się na empirycznym potwierdzeniu wzrostu wydajności uzyskanej dzięki programowaniu równoległemu i10-rdzeniowym procesorom Westmere-EX. Badaniom objęto cztery klasy algorytmów: czysto matematyczny dotyczący problemu Collatza, kryptograficzny 3DES, kwantowy algorytm Grovera oraz klasyczny algorytm genetyczny. Dla zastosowań edukacyjnych dostęp do laboratorium jest bezpłatny, a udostępniane platformy wspierają wszelkie zaawansowane technologie.

Słowa kluczowe: wirtualne labolatorium, wirtualny eksperyment, programowanie równoległe, Manycore Testing Lab

Abstract. The modern didactic process of programming techniques often requires access to the modern hardware and software. In a particular part applies to parallel algorithms, where appropriate properties to a much greater extent can be seen in the new generation of high-performance processors. To create an international academic community associated with this specialization, Intel released a virtual test lab (Manycore Testing Lab - MTL). The paper presents the architectural aspect and the practical application of MTL at work reusable and focuses on empirical confirmation gains obtained through parallel programming and 10-core Westmere-EX processors. The study consisted of four classes of algorithms: for a purely mathematical problem Collatz, 3DES cryptography, quantum Grover algorithm and the classic genetic algorithm. For educational access to the laboratory is free and available to all platforms support advanced technologies.

Key words: virtual labolatory, virtual experiments, parallel programming, Manycore Testing Lab

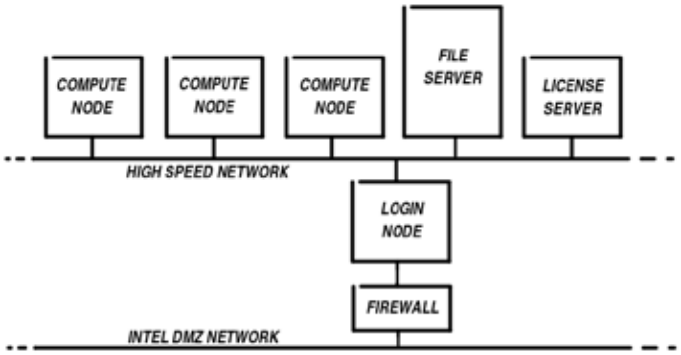
1. Wprowadzenie

Z dostępu do Manycore Testing Lab [1] korzysta ponad 160 uniwersytetów na całym świecie. Intel udostępnia wirtualne laboratoria działające w oparciu o różne systemy operacyjne: Linux lub Windows. Inicjatywa spotkała się z bardzo dużym zainteresowaniem także ze względu na dostęp do nowoczesnego środowiska programistycznego Parallel Studio ze wsparciem najnowszych kompilatorów zoptymalizowanych pod kątem procesorów Intel. Przejaw aktywności w świecie akademickim firmy Intel nie jest nowością. Już wcześniej korporacja nVidia umożliwiła nawiązywanie partnerstw z ośrodkami edukacyjnymi w ramach kształcenia studentów z wykorzystaniem programowalnych akceleratorów graficznych CUDA [14]. Idea nauczania wspierana przez międzynarodowe firmy może wskazywać gałęzie nauki i przemysłu, które będą się bardzo dynamicznie rozwijać.

Intel MTL cechuje budowa wielowęzłowa z wyróżnieniem charakterystycznych nodów logowania (węzeł dostępowy), obliczeniowych, magazynujących informację oraz odpowiedzialnych za zarządzanie licencjami oprogramowania. Schemat budowy systemu został przedstawiony na Ryc. 1.

Węzły obliczeniowe oraz węzeł dostępowy opierają się na platformie zbudowanej z czterech procesorów Intel Xeon E7-4860. Każdy z układów charakteryzuje się częstotliwością taktowania wynoszącą 2.26 GHz, która dzięki technologii Turbo Boost [12] może zostać maksymalnie podniesiona do 2.66 GHz. Pojedynczy procesor posiada 10 rdzeni, jednak za pomocą technologii Hyper-Threading [13] jest możliwa obsługa jednocześnie aż 20 wirtualnych wątków. Dla całej platformy złożonej z czterech procesorów możemy więc uzyskać równoległą obsługę aż 80 wątków. Należy jednak pamiętać, że technologia HT jedynie symuluje działanie tak dużej ilości procesorów. W

1 Instytut Informatyki i Automatyki, Państwowa Wyższa Szkoła Informatyki i Przedsiębiorczości w Łomży, lswierczewski@pwsip.edu.pl



Ryc. 1. Schemat budowy Intel Manycore Testing Lab.
Ryc. 1. Diagram of the Intel Manycore Testing Lab.

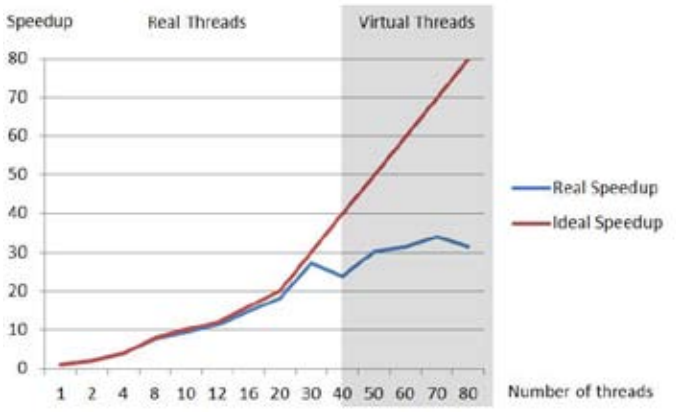
praktyce uzyskany wzrost wydajności pomiędzy 40, a 80 wątkami na tego typu platformie może być minimalny lub nawet w skrajnych przypadkach ujemny. Całość uzupełnia 64 lub 256 GB (w zależności od rodzaju węzła) szybkiej pamięci operacyjnej DDR3 o częstotliwości taktowania 1066 MHz.
Na węzłach zainstalowane są systemy operacyjne Windows lub Linux. Wszystkie obliczenia zaprezentowane w tym artykule wykonano na węźle działającym pod kontrolą systemu operacyjnego Linux i wyposażonym w 256 GB pamięci operacyjnej.
Podczas analiz algorytmów wykorzystano autorską bibliotekę Olib [2] [3]. Zostały w niej zaimplementowane wydajne algorytmy z wykorzystaniem środowiska OpenMP [4], a także nVidia CUDA. Intel MTL wspiera jedynie oprogramowanie pisane z myślą o komputerach równoległych z pamięcią wspólną więc w pracy zostanie wykorzystane jedynie środowisko OpenMP.

2. Analizy wydajnościowe Intel MTL

Pierwszym algorytmem jaki wzięto pod uwagę podczas testowania możliwości wirtualnego laboratorium firmy Intel jest słynny problem z teorii liczb: hipoteza Collatza [5]. Mówi ona, że zaczynając od dowolnej liczby naturalnej i generując kolejne wartości ciągu za pomocą wzoru (1) zawsze dojdziemy do liczby 1.

$$c_{n+1} = \begin{cases} \frac{1}{2} c_n & gdy\ c_n\ jest\ parzysta \\ 3c_n + 1 & gdy\ c_n\ jest\ nieparzysta \end{cases} \qquad (1)$$

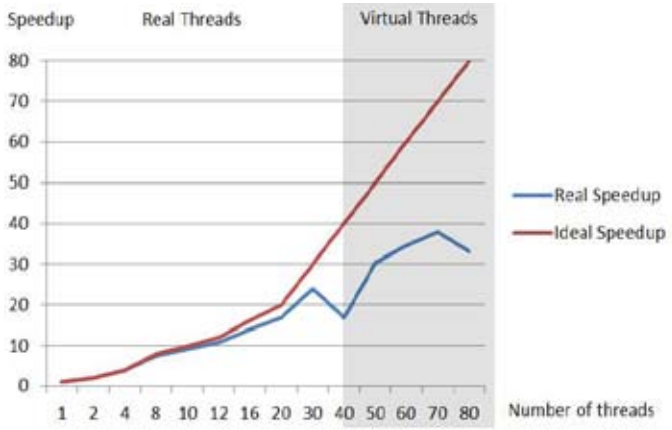
Problem nie został do dnia dzisiejszego rozwiązany pomimo, że wykonano wiele analiz komputerowych [6] [7]. Wyniki zrównoleglenia dla algorytmu Collatza przedstawiono graficznie na Ryc. 2. Bardziej szczegółowe dane można odnaleźć w Tab. 1.



Ryc. 2. Przedstawienie wyników równoleglizacji dla problemu Collatz'a oraz platformy Intel MTL
Ryc. 2. The presentation of the results for the Collatz conjecture and Intel MTL platform parallelization problem

Tab. 1. Tabela przedstawiająca dokładne czasy wykonywania algorytmu rozwiązującego problem Collatz'a dla różnych ilości wątków na platformie Intel MTL
Tab. 1. Table which shows the exact execution time of the algorithm solving the Collatz conjecture for a different number of threads on Intel MTL platform

Number of threads	Time [in seconds]	Real Speedup
1	782,57	1
2	391,933	1,996693312
4	198,4	3,944405242
8	102,235	7,65461926
10	83,655	9,354730739
12	69,109	11,32370603
16	52,953	14,77857723
20	43,05	18,17816492
30	28,71	27,25774991
40	32,749	23,89599682
50	25,882	30,2360714
60	24,866	31,47148717
70	23,054	33,94508545
80	24,902	31,42598988



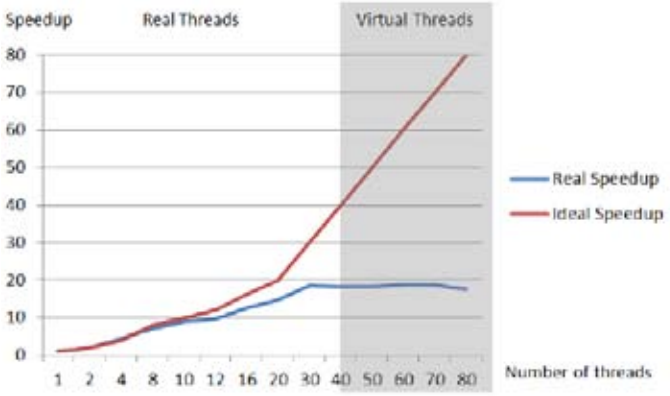
Ryc. 3. Przedstawienie wyników równoleglizacji dla algorytmu 3DES oraz platformy Intel MTL
Ryc. 3. The presentation of the results of the 3DES algorithm and Intel MTL platform parallelization

Kolejnym rodzajem algorytmu był Triple-DES [8]. W bibliotece Olib można odnaleźć implementację 3DES ECB umożliwiającą szyfrowanie bloków o rozmiarze 64-bit w sposób równoległy i całkowicie niezależny na wielu procesorach. Szyfrowany plik miał rozmiar 512 MB. Rezultaty zrównoleglenia kodu i rzeczywiste przyśpieszenie można zaobserwować na Ryc. 3. Poszczególne pomiary czasów realizacji szyfrowania zostały umieszczone w Tab. 2.

Tab. 2. Tabela przedstawiająca dokładne czasy wykonywania algorytmu 3DES dla różnych ilości wątków na platformie Intel MTL
Tab. 2. Table showing the exact execution times of the 3DES algorithm for different number of threads on Intel MTL platform

Number of threads	Time [in seconds]	Real Speedup
1	2729,085	1
2	1374,803	1,985073498
4	710,359	3,841839126
8	361,902	7,540950313
10	298,679	9,137184067
12	253,223	10,77739779
16	196,869	13,86244152
20	161,698	16,87766701
30	113,643	24,01454555
40	161,69	16,87850207
50	89,382	30,5328254
60	79,076	34,51217816
70	72,011	37,89816834
80	82,319	33,15255287

Aktualnie dość młodą i szybko rozwijającą się gałęzią informatyki są algorytmy kwantowe. Nie zbudowano dotychczas w pełni funkcjonalnego komputera kwantowego jednak znamy kilka algorytmów, których działanie można symulować na współczesnych maszynach. Jednym z nich jest algorytm Grovera [9] [10]. Główną operacją która podlega zrównolegleniu podczas symulacji tego algorytmu jest przejście rejestru kubitów przez bramkę kwantową, które jest definiowane jako mnożenie wektora przez macierz. Dla platformy MTL symulację wykonano dla rejestru kwantowego o rozmiarze 16 kubitów co wymusza operowanie na wektorze o długości 2^{16} oraz macierzy o rozmiarze 2^{16} x 2^{16}. Podczas pracy program alokował aż 96 GB pamięci operacyjnej. Wyniki zostały przedstawione na Ryc. 4. Bardziej szczegółowe dane można odnaleźć w Tab. 3.

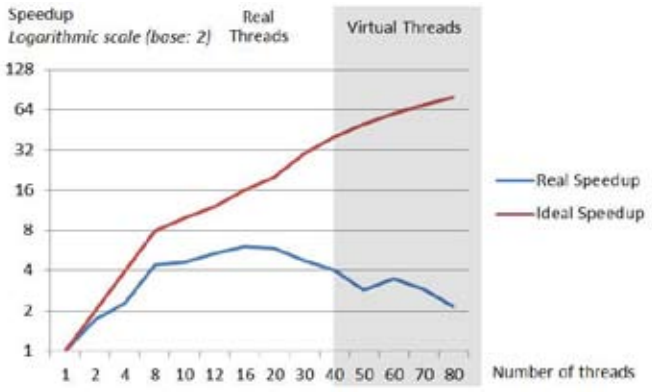


Ryc. 4. Przedstawienie wyników równoleglizacji dla symulacji kwantowego algorytmu Grovera oraz platformy Intel MTL
Ryc. 4. The presentation of the results of simulation quantum Grover's algorithm and Intel MTL platform parallelization

Tab. 3. Tabela przedstawiająca dokładne czasy wykonywania symulacji kwantowego algorytmu Grovera dla różnych ilości wątków na platformie Intel MTL
Tab. 3. Table which shows the exact execution times of simulation quantum Grover's algorithm for different number of threads on Intel MTL platform

Number of threads	Time [in seconds]	Real Speedup
1	7065,58	1
2	3670,555	1,924935058
4	1569,955	4,500498422
8	1017,367	6,944966762
10	792,365	8,917077357
12	728,098	9,704160704
16	565,739	12,48911601
20	484,207	14,59206496
30	384,308	18,38520145
40	386,253	18,29262168
50	387,401	18,23841446
60	376,061	18,78838805
70	378,095	18,68731403
80	400,883	17,62504272

Podczas badań wzięto także pod uwagę potencjalne możliwości zrównoleglenia algorytmu genetycznego [11]. Wykonane pomiary dotyczyły populacji o rozmiarze 16384, krzyżowania typu two-point z prawdopodobieństwem równym 50%. Prawdopodobieństwo zajścia mutacji bit inversion wynosiło 1%. Dodatkowo podczas analiz uwzględniono selekcję za pomocą koła ruletki i skalowanie liniowe. Rezultaty dla algorytmu genetycznego zostały przedstawione w Ryc. 5 i Tab. 4.

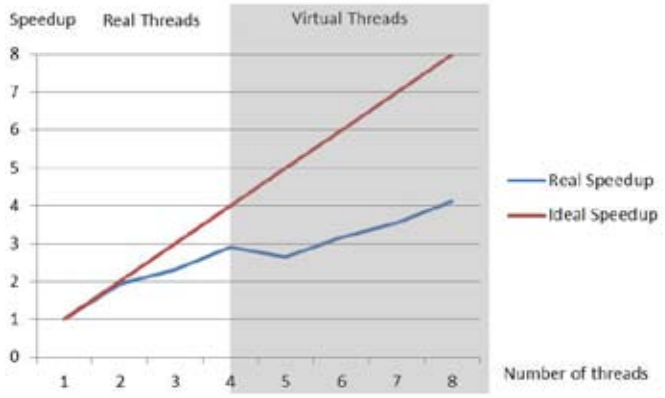


Ryc. 5. Przedstawienie wyników równoleglizacji dla realizacji algorytmu genetycznego oraz platformy Intel MTL
Ryc. 5. The presentation of the results of a genetic algorithm and Intel MTL platform parallelization

Tab. 4. Tabela przedstawiająca dokładne czasy realizacji algorytmu genetycznego dla różnych ilości wątków na platformie Intel MTL
Tab. 4. Table which shows the exact execution times of a genetic algorithm for different number of threads on Intel MTL platform

Number of threads	Time [in seconds]	Real Speedup
1	9074	1
2	5276	1,719863533
4	3976	2,282193159
8	2037	4,454590083
10	1955	4,641432225
12	1678	5,407628129
16	1498	6,05740988
20	1539	5,896036387
30	1902	4,770767613
40	2241	4,04908523
50	3162	2,86970272
60	2601	3,488658208
70	3113	2,914873113
80	4218	2,15125652

Dla uzyskania dodatkowego punktu odniesienia względem innej platformy sprzętowej testy wydajności algorytmu genetycznego wykonano także na komputerze wyposażonym w jeden procesor Intel i7 920 i 24 GB pamięci operacyjnej DDR3. Wszystkie parametry algorytmu genetycznego pozostały takie same jak w przypadku platformy Intel MTL i czterech procesorów opartych o rdzeń Westmere-EX. Wyniki dla dodatkowej platformy zostały zaprezentowane na Ryc. 6 i Tab. 5.



Ryc. 6. Przedstawienie wyników równoleglizacji dla algorytmu genetycznego oraz platformy sprzętowej opartej o procesor Intel i7 920
Ryc. 6. The presentation of the results of a genetic algorithm and hardware platform based on the Intel i7 920 processor parallelization

Tab. 5. Tabela przedstawiająca dokładne czasy wykonywania algorytmu genetycznego dla różnych ilości wątków na platformie sprzętowej opartej o procesor Intel i7 920
Tab. 5. Table which shows the exact execution times of a genetic algorithm for different number of threads on the hardware platform based on the Intel i7 920 processor

Number of threads	Time [in seconds]	Real Speedup
1	7276	1
2	3753	1,938715694
3	3134	2,321633695
4	2507	2,902273634
5	2762	2,634322954
6	2294	3,171752398
7	2057	3,537190083
8	1767	4,117713639

3. Analiza możliwości równoleglizacji na platformie Intel MTL

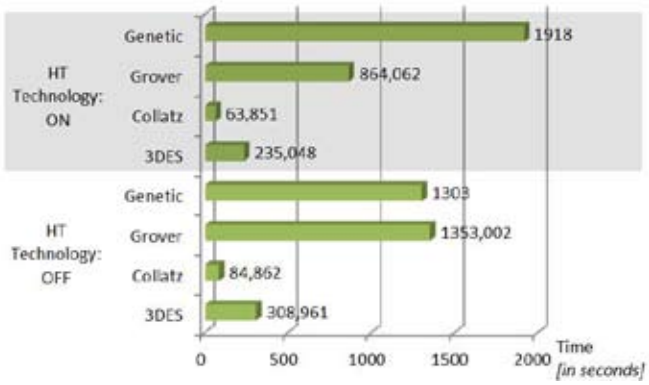
Najwyższe przyśpieszenia uzyskano podczas zrównoleglenia algorytmu 3DES i programu odpowiedzialnego za generowanie ciągu liczb według formuły Collatza. W pierwszym przypadku przyśpieszenie równe ok. 37,89 wygenerowano podczas wykorzystania 70 wątków. W drugim przypadku maksymalne odnotowane przyśpieszenie wyniosło ok. 33,94. Rezultat ten uzyskano także podczas intensywnego korzystania z 70 wątków. Wyniki te świadczą o potencjale jaki posiada technologia Hyper-Threading. Pomimo, że platforma posiada fizycznie 40 procesorów to intuicyjne uruchomienie programu z wykorzystaniem 40 wątków, daje gorsze wyniki niż w przypadku 70 wątków. Należy jednak pamiętać, że w tradycyjnym przypadku uruchomienie większej ilości wątków niż mamy dostępnych procesorów musi doprowadzić do ich przełączania między jednostkami obliczeniowymi przez planistę systemowego co zawsze skutkuje tylko spadkiem wydajności.

Uzyskane rezultaty nie wyglądają jednak tak dobrze w przypadku symulacji kwantowego algorytmu Grovera i algorytmu genetycznego. Dla algorytmu Grovera przyśpieszenie wynoszące 18,78 uzyskano dla 60 wątków. Łatwo jednak zauważyć, że różnice między wynikami czasowymi dla przedziału od 30 do 70 wątków są minimalne. Na zaawansowanej platformie Intel MTL algorytm zachowuje się dość dziwnie. Przyśpieszenie osiąga granicę już przy 16 wątkach z wartością zaledwie 6,05. Testy algorytmu z tego powodu przeprowadzono także na znacznie prostszej konfiguracji sprzętowej dostępnej dla przeciętnego użytkownika. Wyniki dotyczące procesora Intel i 7 920 ukazują, że podczas wykorzystania platformy złożonej tylko z jednego wielordzeniowego CPU trend wzrostu wydajności jest prawidłowy – najlepsze przyśpieszenie rzędu 4,11 uzyskano przy wykorzystaniu wszystkich 8 wątków. Tak mało efektywne działanie algorytmu genetycznego może być spowodowane dość mało efektywną implementacją tego rozwiązania w bibliotece Olib.

Implementacja ta może okazać się wystarczająca dla komputerów, które posiadamy w domu jednak nie wykorzysta w pełni możliwości zaawansowanych i drogich platform jaką jest Intel MTL.

4. Wpływ technologii hyper-threading na czasy realizacji zadań

Podczas analiz sprawdzono także dokładnie wpływ technologii Hyper-Threading na wydajność. Firma Intel nie umożliwia użytkownikowi wyłączenia tej technologii z poziomu BIOSu. Ograniczono się więc do wykorzystania programu taskset dostępnego w środowisku systemu operacyjnego Linux. Umożliwia on zdefiniowanie na stałe z jakich procesorów dostępnych w systemie ma korzystać uruchamiany program. Na platformie złożonej z czterech procesorów posiadających fizycznie 10 rdzeni i wirtualizację 20 wątków uruchomiano oprogramowanie tak aby korzystało tylko z 10 wątków na procesor (w sumie 40 wątków dla całej maszyny). Takie ograniczenie powinno dać podobne wyniki jak wyłączenie wsparcia dla HT z poziomu BIOSu.
Wyniki takiego eksperymentu zaprezentowano na Ryc. 6. Jedynie algorytm genetyczny działał gorzej podczas symulowanego wyłączenia technologii Hyper-Threading. Wszystkie pozostałe algorytmy wykonywały się zauważalnie szybciej dzięki zastosowaniu HT.



Ryc. 6. Porównanie czasów realizacji przedstawionych algorytmów symulacją nieaktywnej i aktywną technologią Hyper-Threading na Intel MTL

Ryc. 6. The comparison of execution times of a showed algorithms using inactive and active Hyper-Threading technology on Intel MTL

5. Wnioski końcowe

Zaprezentowane wyniki świadczą o dużych możliwościach wirtualnego laboratorium Intela. Jedynie działanie algorytmu genetycznego pozostawia wiele do życzenia. Algorytm 3DES z przyśpieszeniem w granicach 37,89 niewiele odbiega od zrównoleglenia doskonałego, dzięki któremu powinniśmy w tym przypadku uzyskać przyśpieszenie równe 40. Wysoce prawdopodobne, że problem z algorytmem genetycznym leży nie w wąskim gardle platformy, a w samym kodzie oprogramowania.

Laboratorium z całą pewnością umożliwia efektywną wizualizację możliwości programowania równoległego i nowoczesnych procesorów wielordzeniowych. Dostęp do MTL w trakcie kursów z technik programowania równoległego może znacząco podnieść ich atrakcyjność.

Literatura (References)

[1] Intel Manycore Testing Lab, URL: http://software.intel.com/en-us/articles/intel-many-core-testing-lab/ [ostatni dostęp: 10.07.2012 r.]
[2] Olib Library, URL: http://goldbach.pl/olib/ [ostatni dostęp: 10.07.2012 r.]
[3] Ł. Świerczewski OLib - High Performance Library Containing Mathematical Functions & Algorithms. International Supercomputing Conference, Hamburg 2012.
[4] R. ChandrA, R. Menon, L. Dagum, D. Kohr, D. Maydan, J. McDonald, Parallel Programming in OpenMP. Morgan Kaufmann 2000.
[5] R. K. Guy, Unsolved problems in Number Theory. Springer (2004): 336–337.
[6] J. Simons, B. De Weger, Theoretical and computational bounds for m-cycles of the 3n + 1 problem, Acta Arithmetica 2005.
[7] Tomas Oliveira e Silva, Computational verification of the 3x+1 conjecture, URL: http://www.ieeta.pt/~tos/3x+1.html, [ostatni dostęp: 10.07.2012 r.]
[8] U.S. DEPARTMENT OF COMMERCE / NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY: (1977) Data Encryption Standard (DES), Federal information processing standards publication
[9] L. K. Grover, A fast quantum mechanical algorithm for database search, Proceedings, 28th Annual ACM Symposium on the Theory of Computing, (1996): 212.
[10] M.A. Nielsen, I.L. Chuang, Quantum computation and quantum information. Cambridge University Press 2000, Chapter 6.
[11] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag 1999.
[12] J. Charles, P. Jassi, N.S. Ananth, A. Sadat, A. Fedorova, Evaluation of the Intel® Core™ i7 Turbo Boost feature, Workload Characterization, 2009. IISWC 2009. IEEE International Symposium
[13] J.R. Bulpin, A.I. Pratt, Multiprogramming performance of the Pentium 4 with Hyper-Threading, In the Third Annual Workshop on Duplicating, Deconstructing and Debunking (WDDD2004) held at ISCA ‘04. (2004): 53-62.
[14] D. Krik, NVIDIA CUDA software and GPU parallel computing architecture, ISMM 2007.