

Raport Științific și Tehnic

Contract nr. 263CI/2018

Cod proiect: PN-III-P2-2.1-CI-2018-1518

Titlu: "Dezvoltarea unor noi algoritmi de optimizare în sisteme de banking"

Durată proiect: 4 luni

1. Descrierea științifică și tehnică a activităților.

În termeni simpli, scopul proiectului este implementarea de software și algoritmi pentru rezolvarea problemei transportului de bunuri (lichidități, documente etc.) între bănci cu scopul minimizării costurilor aferente și satisfacerea cerințelor clienților (bănci).

Descrierea problemei:

- Printre scopurile probleme se regăsesc programarea:
 1. Transportului de bani, de aur, de dosare de credit, precum și de alte materiale necesare funcționării unităților bancare precum și agenților economici, unităților administrative ale statului;
 2. Alimentări și depanări de ATM-uri.

- Obiectivele problemei:
 1. Reducerea timpului de transport, a drumului parcurs, în vederea încadrării în intervalul orar stabilit de client (acesta este obiectivul principal);
 2. Prezentarea de soluții care minimizează numărul de rute (sau autovehicule);
 3. Prezentarea de soluții care minimizează distanța totală parcursă;
 4. Prezentarea de soluții care minimizează timpul de așteptare al autovehiculelor;
 5. Respectarea ordinii cererilor (dacă se cere, de exemplu, transportarea de bunuri de la punctul A la punctul B, atunci nu putem prezenta soluții în care în ruta autovehiculelor se vizitează întâi punctul B și apoi A).

- Constrângeri:
 1. Un interval de timp obligatoriu pentru satisfacerea fiecărei cereri;
 2. Durată specifică pentru satisfacerea fiecărei cereri;

3. Respectarea întoarcerii fiecărei unități de transport centrului de care aparține.
4. Respectarea caracteristicilor de volum și greutate pentru unitățile de transport;
5. Respectarea capacității unităților de transport;
6. Respectarea condițiilor de asigurare ale unităților de transport.

Din punct de vedere tehnic, problema anterior precizată admite diverse formulări și modele informatice, fiecare cu avantaje și dezavantaje, din punct de vedere al rezolvării. Independent de aceste reprezentări ale problemei pe care dorim s-o rezolvăm, cert este că aceasta se încadrează într-o categorie de probleme dificil de rezolvat din punct de vedere al studiului complexității algoritmilor (este vorba de așa numitele probleme NP-hard, subiect studiat îndelung în domeniul Informaticii Teoretice). Știm că problema se încadrează în această categorie de probleme dificile, întrucât există versiuni mult simplificate ale acesteia care au fost abordate în trecut și sunt dovedite a fi dificile (NP-hard). Problema noastră are în plus multe alte constrângeri față de aceste variante simplificate, ceea ce conferă caracterul de noutate și inovare al abordării noastre. În continuare vom prezenta o scurtă descriere a domeniului din informatică ce se ocupă de problemele dificile (NP-hard).

Scurta prezentare a problemelor NP-hard

Dispozitivele computerizate utilizate în zilele noastre într-o varietate de aplicații complexe pot efectua miliarde de operațiuni de bază pe secundă. Cu toate acestea, există multe probleme importante care necesită un timp de rezolvare semnificativ, chiar și în cazul celor mai rapide computere. În vederea înțelegerii dificultăților diferitelor probleme de calcul, acestea au fost împărțite în clase de complexitate.

O clasă de complexitate fundamentală este clasa P. Aceasta clasă constă din acele probleme care sunt solvabile în timp polinomial. Aceste probleme sunt de obicei *rezolvabile* în practică.

Nu toate problemele sunt însă în clasa P: algoritmi mai rapizi cunoscuți pentru rezolvarea unor astfel de probleme necesită o căutare combinatorială pentru care timpul de rulare poate crește exponențial cu lungimea intrării. Aceste probleme nu sunt artificiale; ele au aplicații importante în multe domenii: bioinformatică, transporturi, cercetări operaționale, medicină, rețele etc.. Problemele care par să aibă doar soluții în timp exponențial sunt grupate folosind clasa NP (probleme rezolvabile prin algoritmi nedeterminiști polinomiali).

Clasa NP constă în acele probleme ale căror soluții care pot fi verificate în timp polinomial. NP este clasa problemelor de decizie pentru care este ușor de verificat corectitudinea unui răspuns, cu ajutorul unei cantități reduse de informații, dar acest lucru nu înseamnă găsirea unei soluții, ci doar verificarea dacă o presupusă soluție este corectă. Fiecare problemă din această clasă poate fi rezolvată în

timp exponențial folosind o căutare exhaustivă. O problemă este NP-hard dacă toate problemele din NP sunt reductibile la ea în timp polinomial.

Dacă există un algoritm de timp polinomial pentru oricare dintre aceste probleme, toate problemele în NP ar fi rezolvabile în timp polinomial. Conceptul de problemă NP-hard este fundamental în informatica teoretică atât din motive teoretice cât și practice.

Asa cum a fost prezentat la începutul raportului, problema pe care o studiem se încadrează în clasa problemelor NP-hard. Așadar, din cauză că aceste probleme sunt intrinsec dificil de rezolvat, există mai multe metode care se pot utiliza pentru a produce un algoritm utilizabil în practică. Întrucât noi dorim rezolvarea din punct de vedere practic a problemei, zona noastră de interes este în zona algoritmilor euristici. În acest sens, implementăm algoritmi bazați pe tehnici din zona de bioinformatică, anume algoritmi genetici.

Algoritmi genetici

Algoritmii genetici sunt inspirați din recombinația cromozomilor ADN și urmăresc evoluția unei populații de „indivizi” care codifică soluții ale problemei de-a lungul a mai multe generații (iterații ale algoritmului). Această metodă este de asemenea îndelung studiată și documentată ca și euristică evoluționistă, dar aplicarea ei la problema noastră nu este imediată, ci necesită experimentare și reglare.

În cadrul algoritmilor genetici se alege o „populație” inițială compusă din reprezentări binare numite cromozomi care sunt inspirate din biologie (structura ADN). Acești cromozomi codifică soluții ale problemei (aici soluția este alcătuită din trasee ale autovehiculelor). Pentru fiecare soluție vom avea o reprezentare binară sub formă de cromozom, iar pentru fiecare cromozom vom putea obține soluția care îi corespunde, altfel spus, avem nevoie de o funcție bijectivă care transformă (codifică) soluții în reprezentarea binară cromozomală.

Metoda de alegere a populației inițiale de cromozomi poate să fie bazată pe algoritmi simpli (de exemplu se poate alege aleator câte o soluție din spațiul soluțiilor fezabile). După acest prim pas, populația trece prin anumite etape sau iterații ale algoritmului care au scopul de a rafina soluțiile prin procese inspirate din procese evoluționiste. Mai exact, este vorba despre crossover, mutație și evaluarea parametrului fitness.

Întâi, crossover-ul este o operație ce provine din recombinația ADN, în care participă doi cromozomi: se alege un punct (sau mai multe) de rupere al cromozomilor, după care acești cromozomi schimbă între ei bucățile de după punctul de rupere. Astfel putem obține noi indivizi care au caracteristici provenite din ambii indivizi generatori. Acest lucru este permis de faptul că fiecare cromozom este o reprezentare binară ce codifică o soluție a problemei.

Mutația, pe de altă parte, este o operație aleatoare prin care e alterează local forma binară a fiecărui cromozom cu o șansă mică pentru fiecare bit component. Această etapă are rolul de a împiedica stagnarea populației: dacă indivizii devin prea asemănători, crossover-ul va crea tot indivizi asemănători, dar mutația permite populației să continue să încerce noi soluții.

În final, avem o măsură a fitness-ului fiecărui cromozom. Această măsură exprimă cât de eficientă este soluția corespunzătoare fiecărui cromozom. În cazul nostru, această funcție exprimă minimizarea costurilor asociate transportării prin intermediul flotei de autovehicule și ia în considerare toate celelalte constrângeri menționate anterior. Ne dorim ca soluțiile bune să determine o valoare de fitness mai mare pentru cromozomii asociați. Aceasta ne va permite să implementăm o etapă de selecție în care păstrăm pentru generațiile următoare indivizii care au fitness mai mare.

Pe scurt, nucleul acestei tehnici algoritmice este următorul: populația trece prin etapa de crossover în care se aleg diverse perechi de cromozomi care produc noi indivizi. Apoi, se trece la etapa de mutație în care se variază local cromozomii pentru a preveni stagnarea. În cele din urmă, se calculează fitness-ul fiecărui cromozom și se sortează populația după fitness. O parte din populație (partea care are fitness mic) va dispărea din populația curentă în etapa de selecție.

Tehnica menționată mai sus permite ca să putem testa soluții din ce în ce mai bune. La un moment dat, când nu se mai poate îmbunătăți cea mai buna soluție găsită până la momentul respectiv, algoritmul se oprește și prezintă soluția. Această euristică prezintă rezultate foarte bune în practică, cu toate că și în zilele noastre se fac eforturi pentru a demonstra convergența acestui algoritm la soluția optimă (însă acest fapt este comun majorității tehnicilor euristice pentru rezolvarea problemelor dificile).

Tehnologii folosite

Pe lângă partea teoretică, eforturi deosebite au fost necesare la partea de implementare. În continuare vom prezenta pe scurt, tehnologiile folosite și detaliile tehnice.

Aplicația este structurată pe 3 componente principale:

- Componenta de permanentizare a datelor (baza de date)
- Componenta de logică
- Componenta de interfață cu utilizatorul

Tehnologia în care a fost realizată implementarea este reprezentată de limbajul de programare C# 6.0, framework-ul .NET 4.6.2.

Pentru componenta de permanentizare a datelor s-a considerat utilizarea unui ORM (Object Relational Mapper) sub tehnologia EntityFramework 6.0, din cadrul framework-ului .NET. Aspectul inovativ al acestei tehnologii consta în faptul ca nu este necesara dezvoltarea de cod SQL sau de proceduri stocate pentru a realiza interogări asupra bazei de date, ci se utilizează direct LINQ (Language-integrated-Query) pentru realizarea acestor interacțiuni. Corespondența dintre codul C# și tabelele din baza de date se realizează prin intermediul unor clase care poarta denumirea de entități. Prin intermediul componentei ORM, codul C# este translatat automat în statement-uri SQL (interogări, selecții, inserări, ștergeri, modificări). Pentru baza de date s-a considerat RDBMS-ul (sistemul de management al bazelor de date) furnizat de compania Oracle, versiunea 12 C. Conectorul de integrare cu baza de date este librăria Devart, versiunea 5.0. Există un număr de 15 entități care se translatează în tabele la nivelul bazei de date.

Pentru componenta de logică s-a considerat structurarea funcționalităților pe clase de tip "Manager", pentru a particulariza responsabilitățile. În dezvoltarea subcomponentelor de logică s-au respectat principiile SOLID.

Una din cele mai importante clase dezvoltate este "OptimizationManager", în cadrul căreia sunt implementați algoritmi genetici pentru optimizarea funcției de cost. Implementările realizate în cadrul acestei clase se bazează pe tehnologia Task Parallel Library, disponibilă în cadrul frameworkului .NET. Scopul acestei componente este de a simplifica procesul de paralelizare și de rulare concurentă a firelor de execuție în cadrul evoluției algoritmilor genetici. Aceasta librărie optimizează modul de folosire al tuturor procesoarelor disponibile pe sistemul hardware pe care rulează aplicația. De asemenea, aceasta partiționează foarte eficient unitățile de munca distribuite către procese, organizează firele de execuție, gestionează stările proceselor, disponibilizează și recuperează resursele.

Pentru a realiza logarea tuturor acțiunilor realizate la nivelul aplicației, în cadrul componentei de logică s-a utilizat librăria Log4Net.

S-a realizat integrarea cu API-ul de geolocație Google Maps, pentru descărcarea informațiilor privind punctele de atins, pentru descărcarea informațiilor de distanță între punctele din trafic și pentru reprezentarea traseelor mașinilor pe hartă.

Cea de-a treia componentă a aplicației, interfața cu utilizatorul, este construită pe baza limbajului C# .NET și a tehnologiilor ASP.NET WebForms, ASP.NET MVC, precum și a framework-ului JQuery bazat pe Javascript. Aceasta componentă este reprezentată de o aplicație web. Pentru design-ul aplicației s-a utilizat tehnologia Bootstrap, după cele mai recente standarde recomandate de companii precum Microsoft și Google. Reprezentarea informațiilor pe hartă (puncte de destinație, trasee) se realizează prin interogarea API-urilor Google.

2. Punerea în evidență a modului de valorificare și a rezultatelor cecului de inovare.

Firma Holistion Software Consulting SRL realizează activități de dezvoltare de software în calitate de subcontractor al companiei SITI SA pentru Banca Comercială Română. Principalul proiect la care se realizează dezvoltări este pentru subsidiara CIT ONE a BCR, subsidiara a cărei misiune este de a realiza transportul de numerar, lichidități, aur, documente bancare, intervenții pentru ATM-uri. Holistion asigură mentenanța și dezvoltările necesare pentru platforma software de logistică a companiei CIT ONE.

În cadrul platformei software sunt definite toate resursele precum și toate obiectivele companiei CIT ONE. Spre exemplu, în cadrul resurselor se pot menționa următoarele: centre de procesare, angajați cu diferite responsabilități (casieri, trezoreri, șoferi, experți backoffice), unități de transport (blindate, mașini de intervenție pentru depanare ATM), seif-uri, aparatura pentru verificarea și procesarea numerarului. Obiectivele companiei CIT ONE constau în realizarea comenzilor solicitate de clienții care pot fi reprezentați de: unități bancare (din rețeaua BCR, Raiffeisen, BRD, Banca Românească, Banca Transilvania, ING, UniCredit), agenți economici (agenții ale OMV, Mol, Auchan, Megaimage, etc.), instituții ale Statului Român (ANAF), unități ale Băncii Naționale a României. Comenzile realizate de clienți pot fi din diverse categorii precum: cereri de transport numerar (colectări și remiteri de numerar), cereri de ATM-uri (alimentări și goliri de ATM-uri), cereri de transport de aur, cereri de materiale (documente bancare), cereri de intervenții în vederea depanării ATM-urilor (pe diverse categorii de incidente FLM, SLM, TLM), transbordări cu alți operatori transportatori de cash.

Natura activității companiei CIT ONE este deosebit de complexă având în vedere diversitatea resurselor avute la dispoziție precum și variabilitatea comenzilor de realizat.

În acest sens, compania Holistion va propune un produs software pentru optimizarea rutelor de transport realizate de CIT ONE. Acest produs va genera o reducere estimată între 10% și 20% a costurilor pentru compania client prin utilizarea optimă a resurselor disponibile. Un alt aspect important este creșterea satisfacției clienților, având în vedere faptul că produsul software va genera o respectare mai precisă a intervalelor de timp în cadrul cărora aceștia doresc să le fie realizată comanda (Service Level Agreement). Respectarea intervalelor de timp pentru realizarea unei comenzi va genera totodată o reducere a costurilor de penalizare pentru compania CIT ONE.

Prin intermediul cecului de inovare se va realiza un set de algoritmi euristici de optimizare pentru generarea unor rute de transport optime. Acești algoritmi euristici vor fi implementați în cadrul unei aplicații software prototip.

Astfel, aceasta aplicație va fi valorificată prin vânzare către clientul CIT ONE prin intermediul companiei SITI.

Întrucât pe piață exista companii cu business-uri asemănătoare în domeniul logisticii, aplicația poate fi adaptată pentru a răspunde și nevoilor acestor companii. Astfel, potențialul de valorificare al aplicației nu este doar pentru un singur client.

3. Gradul de realizare a rezultatelor estimate în ofertă.

În propunerea de proiect am menționat următoarele livrabile, care sunt realizate în cea mai mare parte.

1. *Crearea unui nou modul software pentru eficientizarea rutelor de transport, modul bazat pe algoritmi euristici de optimizare.*

Modulul software este completat în proporție de 90%: este funcțional, dar mai sunt necesare câteva ajustări, în special la partea de interfață. Ne propunem să finalizăm modulul în următoarele 2 săptămâni.

2. Manual de utilizare al modulului software.

Acest manual este completat în proporție de 100%.

3. Raport tehnic în care se vor descrie algoritmii folosiți.

Pana în prezent ne-am concentrat pe dezvoltarea aplicației. De aceea, raportul tehnic este realizat în proporție de 50%.

Raportare indicatorii de stare și de progres realizați.

- documente contractuale încheiate cu clienții actuali;

Am atașat o copie după contractul cu compania SITI.

- rezultatele exercițiilor financiare ale beneficiarului;

Pentru rezultatele exercițiilor financiare ale beneficiarului s-a anexat bilanța pentru anul 2017. Activitatea firmei a început efectiv spre finalul anului 2017.

- documente de instruire a utilizatorilor;

S-a atașat manualul de utilizare al aplicației.

**Contractor - Furnizor de servicii
Rector**

Mircea Dumitru
Semnătura
Ștampila

**Director
Administrativ Adjunct**

Aida Alexandroaia
Semnătura

**General
Responsabil de Proiect**

Alexandru Popa
Semnătura

**Avizat, Contractor - Beneficiar Reprezentant
Legal**

Lucian-Ionuț Gavrilă
Semnătura
Ștampila

Contabil

Viorica Boeriu
Semnătura

Responsabil de Proiect

Lucian-Ionuț Gavrilă
Semnătura