Simpozijum MATEMATIKA I PRIMENE, Matematički fakultet, Univerzitet u Beogradu, 2016 ,Vol. VII(1)

Problem raspoređivanja vozila pri optimizaciji troškova transporta poljoprivrednih sirovina

Ana Anokić

Faculty of Agriculture, University of Belgrade, Nemanjina 6, 11 080 Zemun, Belgrade, Serbia
e-mail: anokicana@agrif.bg.ac.rs

Zorica Stanimirović

Faculty of Mathematics, University of Belgrade, Studentski trg 16/IV, 11 000 Belgrade, Serbia
e-mail: zoricast@matf.bg.ac.rs

Đorđe Stakić

Faculty of Mathematics, University of Belgrade, Studentski trg 16/IV, 11 000 Belgrade, Serbia
e-mail: djordjes@matf.bg.ac.rs

Tatjana Davidović

Mathematical Institute of the Serbian Academy of Science and Arts, Kneza Mihaila 36, 11000 Belgrade, Serbia
e-mail: tanjad@mi.sanu.ac.rs

Apstrakt. U radu je razmatrana varijanta problema raspoređivanja vozila (engl. *Vehicle Scheduling Problem - VSP*) koja se odnosi na optimizaciju transporta u jednoj fabrici šećera u Srbiji. Predstavljen je precizan matematički model problema VSP koji je oblika mešovitog celobrojnog programiranja sa kvadratnim ograničenjima (engl. *Mixed integer quadratically constrained program - MIQCP*). MIQCP model je korišćen u okviru egzaktnog rešavača Lingo 17 u cilju dobijanja rešenja na skupu realnih i generisanih test primera. Kako su optimalna ili dopustiva rešenja dobijena samo za test primere malih dimenzija, razvijena je Opšta metoda promenljivih okolina (engl. *General Variable Neighborhood Search - GVNS*), kao metaheuristički pristup rešavanju razmatranog problema VSP. Izložene su karakteristike predložene GVNS metode i opisani njeni elementi, prilagođeni karakteristikama posmatrane varijante problema VSP. Analiza eksperimentalnih rezultata pokazuje da predložena GVNS metoda brzo dostiže sva poznata optimalna rešenja na test primerima malih dimenzija i daje rešenja na test primerima srednjih i većih dimenzija za koje optimalno rešenje nije poznato.

Ključne reči: problem raspoređivanje vozila; optimizacija transportnih troškova; MIQCP; metaheuristike, opšta metoda promenljivih okolina.

1. Uvod

Potreba za optimizacijom transporta u brojnim transportnim sistemima prirodno vodi ka velikom interesovanju istraživača za različite varijante problema raspoređivanja vozila (engl. *Vehicle Scheduling Problem - VSP*) sa teorijskog i praktičnog aspekta. Problemi VSP se u opštem slučaju svode na određivanje niza tura koje vozilo realizuje tokom radnog dana sa ciljem smanjenja troškova transporta. Dantzig i Fulkerson su u radu [7] 1954. godine predložili model linearnog programiranja za problem određivanja minimalnog broja tankera u cilju realizacije unapred poznatog plana transporta goriva. Njihov rad podstakao je rad drugih istraživača širom sveta i samim tim i nastanak brojnih varijanti problema raspoređivanja vozila i razvoj metoda za njihovo rešavanje.

Značajan broj varijanti problema VSP odnosi se na optimizaciju prevoza ljudi u urbanim sredinama. Baita i sar. su u studiji [4] za rešavanje problema optimizacije javnog prevoza predložili tri pristupa: *tradicionalni*, zasnovan na skaliranju različitih kriterijuma, *leksikografski*, koji koristi principe logičkog programiranja i *metod Pareto optimizacije* implementiran pomoću genetskih algoritama. Haghani i sar. su u radu [9] analizirali tri modela za raspoređivanja vozila pri autobuskom prevozu putnika. Wang i Shen su u studiji [21] razmatrali problem raspoređivanja električnih autobusa i za njegovo rešavanje primenili algoritam zasnovan na metodi optimizacije kolonijom mrava. Pregled problema raspoređivanja vozila može se pronaći u [2].

Problemi raspoređivanja vozila se često posmatraju zajedno sa problemima rutiranja vozila (engl. *Vehicle Routing Problem - VRP*). Preciznije, za dati skup vozila neophodno je definisati raspored po kojem vozila prave rute, a istovremeno se vrši optimalan izbor lokacija u okviru svake rute. Takvi problemi nazivaju se integrisani problemi rutiranja i raspoređivanja vozila. Moons i sar. su u radu [17] ukazali na značaj ovakvog pristupa, naglasivši da odvojeno posmatranje aspekata raspoređivanja i rutiranja vodi ka suboptimalnim rešenjima problema. Agustina i sar. su u studiji [1] razmatrali integrisan problem raspoređivanja i rutiranja vozila pri transportu hrane, u cilju minimizacije ukupnih troškova, koji uključuju transportne troškove i troškove penala u slučaju ranije ili kasnije dostave od planrane. Metodu simuliranog kaljenja primenili su Xiao i sar. u [22] za rešavanje zelenog problema rutiranja i raspoređivanja vozila. Matematička formulacija problema predloženog u [22] uključuje tri funkcije cilja koje se, redom, odnose na: minimizaciju ukupne CO_2 emisije, ukupno pređeno rastojanje i ukupno vreme putovanja. Androutsopoulos i Zografos su u radu [3] integrisali problem rutiranja i raspoređivanja vozila pri rešavanju realnog problema određivanja ruta opsluživanja i rasporeda putovanja tako da ukupni troškovi transporta i potrošnje goriva budu minimalni, pri čemu su uzeti u obzir i promenljivi uslovi u saobraćaju tokom vremena. Aspekti rutiranja i raspoređivanja integrisani su u algoritmu zasnovanom na optimizaciji mravljim kolonijama predloženom u [3] za rešavanje posmatranog problema.

Organizacija transporta poljoprivrednih dobara ima svoje specifičnosti koje potiču uglavnom od niske cene sirovina na tržištu. S obzirom da troškovi transporta čine glavne troškove proizvodnje, ušteda koja se može ostvariti u ovoj fazi je od velikog značaja. Kako bi motivisali individualne proizvođače da nastave saradnju, kompanije koje se bave otkupom i preradom poljoprivrednih sirovina, organizuju i finansiraju transport od svakog proizvođača do fabrike. Preciznije, kompanije obično iznajmljuju vozila i angažuju radnike koji vrše poslove tranporta, utovara i istovara tokom proizvodne sezone. Iz tog razloga, efikasna organizacija transporta uz ostvarenu uštedu utrošenog vremena i novca je od velikog značaja u preradi poljoprivrednih prozvoda. U literaturi postoji nekoliko radova koji se odnose na razne varijante problema transporta poljoprivrednih sirovina. Higgins je u [11] razmatrao problem raspoređivanja vozila pri transportu šećerne trske u australijskoj fabrici šećera. Autor je predložio model mešovitog celobrojnog linearnog programiranja (engl. *Mixed Integer Linear Program - MILP*) koji opisuje razmatrani problem i za rešavanje realnih test primera razvio dve metaheurističke metode, tabu pretraživanje (engl. *Tabu Search - TS*) i metodu promenljivih okolina (engl. *Variable Neighborhood Search - VNS*). Eksperimentalni rezultati u [11] pokazali su da je su obe dizajnirane metaheuristike uspele da pronađu rešenja sa prosečnim smanjenjem redova čekanja za oko 90% u poređenju sa dotadašnjim metodama primenjivanim u praksi. Milan i sar. su u studiji [14] razmatrali problem transporta secérne trske na Kubi, sa ciljem smanjenja transportnih troškova integrisanjem drumskog i železničkog saobraćaja. Za razmatrani problem predložen je MILP model, koji je potom testiran korišćenjem komercijalnog rešavača HyperLINDO u cilju pronalaženja optimalnih rešenja na realnim test primerima. Kako HyperLINDO nije uspeo da reši ceo model za 200h izvršavanja, autori su najpre rešili potproblem polaznog problema, a zatim iskoristili dobijeno rešenje za konstrukciju dopustivog rešenja polaznog modela. Thuankaewsing i sar. su u radu [18] razmatrali problem transprorta šećerne trske na Tajvanu koji za cilj ima maksimizaciju očekivanih prinosa pod istim uslovima za sve proizvođače.

Predmet ovog rada je problem raspoređivanja vozila pri transportu šećerne repe u jednoj kompaniji u Srbiji. Varijanta razmatranog problema VSP se razlikuje od problema iz literature [11, 14, 18] zbog razlika između šećerne trske i šećerne repe u njihovoj održivosti na otvorenom, tipu korišćenih vozila pri transportu i fabričkih resursa. Razmatrani problem je formulisan u vidu mešovitog celobrojnog programa sa kvadratnim ograničenjima (engl. *Mixed Integer Quadratically Constrained Program - MIQCP*). Razvijeni MIQCP model je korišćen u okviru egzaktnog rešavača Lingo 17 u cilju dobijanja rešenja na skupu realnih i generisanih test primera. Dobijena su optimalna rešenja samo na test primerima manjih dimenzija. Iz tog razloga heuristički pristup predstavlja logičan izbor za rešavanje instanci većih dimenzija.

Metoda promenljivih okolina (VNS) je metaheuristička metoda koja je do sada uspešno primenjena na brojne probleme optimizacije, uključujući i probleme rutiranja i raspoređivanja vozila [10]. Macedo i sar. su u [13] primenili adaptivni metod promenljivih okolina (engl. *Skewed Variable Neighborhood Search - SVNS*) na lokacijski problem rutiranja i raspoređivanja vozila (engl. *Location Routing Scheduling Problem*). De Armas i Melian-Batista su u studiji [8] predložili VNS za rešavanje dinamičkog problema rutiranja vozila sa vremenskim okvirima (engl. *Dynamic Vehicle Routing Problem with Time Windows*). U radu Cheikh i sar. [6], razvijena je VNS metoda za rešavanje problema rutiranja vozila sa višestrukim putovanjima (engl. *Vehicle Routing Problem with Multiple Trips*).

Brojni primeri uspešne primene VNS metode na probleme rutiranja i raspoređivanja vozila bile su motivacija za dizajniranje i implementaciju opšte metode promenljivih okolina (engl. *General Variable Neighborhood Search - GVNS*) za rešavanje razmatranog VSP. Predložena GVNS metoda koristi adekvatne strukture okolina u fazi razmrdavanja, a umesto klasčne lokalne pretrage koristi metodu promenljivog spusta (engl. *Variable Neighborhood Descent - VND*). Eksperimentalni rezultati na realnim instancama manjih dimenzija pokazuju da predložena

GVNS metoda brzo dostiže sva poznata optimalna rešenja dobijena egzaktnim rešavačem Lingo 17, dok je na instancama većih dimenzija, GVNS metoda dostigla ili popravila gornje granice optimalnih rešenja dobijenih Lingo rešavačem za veoma kratko vreme izvršavanja. GVNS se pokazao efikasnim i pri rešavanju generisanih instanci većih dimenzija, koje prate strukturu realnih instanci.

Rad je koncipiran na sledeći način. U sekciji 2 detaljno je izložen razmatrani problem raspoređivanja vozila pri transportu šećerne repe, a odgovarajuci MIQCP model prikazan je u sekciji 3. U sekciji 4 opisana je dizajnirana GVNS metoda za rešavanje razmatranog problema. Eksperimentalni rezultati su izloženi u sekciji 5 i data je njihova analiza. Konačno, u sekciji 6 izvedeni su zaključci i ukazano je na moguće pravce budućih istraživanja.

2. Opis problema

Sezona sakupljanja šećerne repe u Srbiji, tzv. kampanja počinje u septembru i traje uglavnom do decembra ili januara sledeće godine. Okvirni plan transporta šećerne repe definiše se u kasno proleće, kada proizvođači mogu da procene količine šećerne repe koje će spremiti za otkup. Međutim, precizan plan transporta se formira na dnevnom nivou tokom kampanje, tako što se za svaki datum u toku posmatranog perioda unapred priprema lista lokacija koja će biti posećena toga dana. Pritom, veoma važan uslov je da dnevne potrebe fabrike, u smislu količine pristigle robe, moraju biti zadovoljene. Kada se pokrenu fabrička postrojenja, proces proizvodnje ne bi smeo da se zaustavi, s obzirom da ponovno pokretanje mašina stvara značajne dodatne troškove. Takođe, najavljeni vremenski uslovi kao što su velike kiše, mogu primorati proizvođače da pripreme robu nekoliko dana pre dogovorenog datuma. Šećerna repa, izvađena iz zemlje, ne bi trebalo da stoji previše dugo na otvorenom kako ne bi izgubila kvalitet. U slučaju da se nekoj lokaciji repa nalazi duže od dozvoljenog broja dana, posmatrana lokacija se smatra hitnom i mora se isprazniti tokom radnog dana.

Pretpostavka je da su vozila koja se koriste za transport homogena, odnosno imaju isti kapacitet i prosečnu brzinu. Sva vozila su locirana u krugu fabrike, koja je početna i krajnja tačka svake ture. Dnevni raspored vozila definisan je nizom tura i odgovarajućim vremenima polazaka iz kruga fabrike. U svakoj turi, vozilo posećuje samo jednu lokaciju i vraća se u fabriku. Količine pripremljene šećerne repe na svakoj lokaciji značajno su veće od kapaciteta vozila, pa je stoga potrebno svaku lokaciju posetiti nekoliko puta kako bi se ispraznila. Kada se vozilo vrati u krug fabrike, uzimaju se uzorci i vrši se procena kvaliteta transportovane šećerne repe. Vozila ne prave rute kako bi se izbeglo mešanje repe sa različitih lokacija. Čak i u slučaju kada vozilo nije potpuno natovareno (što se može desiti pri poslednjoj poseti lokaciji) ono se vraća u fabriku, opslužujući samo jednu lokaciju. Posle završenog istovara, vozilo može započeti novu turu. Zbog jednostavnosti, pretpostavlja se da postoji dovoljno radne snage i prostora na svakoj lokaciji i u krugu fabrike za istovremeni utovar i istovar neograničenog broja vozila.

Raspored vozila je definisan nizom tura i odgovarajućim vremenima polazaka iz kruga fabrike. Cilj problema je pronaći optimalan raspored vozila koji zadovoljava postavljena ograničenja i pritom minimizira maksimalno radno vreme na skupu posmatranih vozila, odnosno trenutak kada sva vozila završe svoje ture za posmatrani radni dan. U nastavku je dat pregled karakteristika razmatranog problema raspoređivanja vozila pri transportu šećerne repe:

- Vozila su homogena i koriste se više puta tokom radnog dana;
- Krug fabrike predstavlja početnu i krajnju tačku svake ture;
- Svako vozilo u svakoj turi posećuje samo jednu lokaciju;
- Dnevne potrebe fabrike, u smislu količine prevezene robe, moraju biti zadovoljene;
- Postoji gornja granica u broju dana koliko roba može da stoji na otvorenom;
- Svaka lokacija na kojoj roba stoji na otvorenom duže nego što je dozvoljeno mora biti opslužena;
- Cilj je minimizovati vremenski trenutak kada sva vozila završe svoje poslednje ture.

3. MICP model razmatranog problema

Opisani problem je formulisan u obliku mešovitog celobrojnog programa sa kvadratnim ograničenjima (engl. *Mixed integer quadratically constrained problem - MIQCP*). U narednim podsekcijama opisani su datalji MIQCP formulacije razmatranog problema.

3.1. Notacija

Za predstavljanje MIQCP modela uvedena je sledeća notacija:

J : Skup lokacija;
I : Skup vozila;
K : Skup tura;
n : Ukupan broj lokacija;
m : Ukupan broj vozila;
k_{max} : Maksimalan broj tura koje vozilo može da napravi u toku dana;
c_j : Količina sakupljene sirovine na lokaciji $j \in J$;
d_j : Rastojanje od farike do lokacije $j \in J$;
CV: Kapacitet vozila;
C: Dnevne potrebe fabrike;
v: Prosečna brzina vozila;
u: Prosečno vreme zadržavanja u fabrici po završetku ture, potrebno za istovar i analiziranje uzoraka;
w: Prosečno vreme trajanja utovara;
t_j: Broj dana koliko roba stoji na otvorenom na lokaciji $j \in J$;
t_0: Maksimalan broj dana koliko roba može da stoji na otvorenom bez gubitka kvaliteta;
T_j: Binarna veličina dodeljena lokaciji $j \in J$, definisana sa: $T_j = 1$ ako je $t_j > t_0$ i $T_j = 0$ u suprotnom.
ε: Mala pozitivna konstanta, manja od *CV* i c_j/CV, $j \in J$.
t_{start}: Početak radnog vremena;
t_{end}: Kraj radnog vremena;

3.2. Koncept *virtualnih* tura

Različita vozila ne moraju realizovati isti broj tura tokom radnog dana. Broj tura jednog vozila direktno zavisi od udaljenosti lokacija koje to vozilo posećuje od fabrike. Udaljenost direktno određuje vreme trajanja putovanja u oba smera, a samim tim i broj realizovanih tura tokom ograničenog radnog vremena. U cilju jednostavnije formulacije matematičkog modela, uveden je koncept *virtualne ture*, koji ne narušava ograničenja problema. Preciznije, izjednačen je broj tura svih vozila tako što je dopušteno da vozila u svom nizu tura tokom radnog dana imaju jednu ili više virtualnih tura. Pretpostavka je da, u toku virtualne ture, vozilo ostaje u krugu fabrike, tj. ne posećuje ni jednu lokaciju, pa je trajanje virtualne ture jednako nuli. Imajući u vidu da je cilj problema minimizacija maksimalnog od vremena završetaka radnog dana svih vozila, virtualne ture ne utiču na vrednost funkcije cilja, te se zanemaruju pri računanju vremena završetka radnog dana vozila. Zbog jednostavnosti, virtualne ture se nalaze na kraju niza tura koji odgovara vozilu, odmah iza poslednje ture u kojoj vozilo opslužuje neku od lokacija. Dodavanje virtualnih tura, ukoliko je potrebno, omogućava dopunu broja tura svakog vozila do maksimalnih k_{max} bez uticaja na ograničenja problema i na vrednost funkcije cilja.

3.3. Promenljive

MIQCP formulacija koristi sledeće promenljive:

- Binarna promenljiva x_{ik}^j ima vrednost 1 ako vozilo $i \in I$ posećuje lokaciju $j \in J$ u turi $k \in K$, a 0 u suprotnom. Ako je tura $k \in K$ vozila $i \in I$ virtualna, tada je $\sum\limits_{j \in J} x_{ik}^j = 0$;
- Realna promenljiva t_{ik} definiše vreme polaska vozila $i \in I$ u turi $k \in K$ iz kruga fabrike;
- Binarna promenljiva y_j, $j \in J$ je uvedena u cilju praćenja ukupne količine prevezene repe do fabrike sa lokacije $j \in J$. Vrednost promenljive y_j jednaka je 0 ako je lokacija $j \in J$ ispražnjena, a 1 u suprotnom. Ako je $y_j = 1$, količina šećerne repe koja je transportovana sa lokacije $j \in J$ u fabriku jednaka je c_j. U suprotnom, količina robe dovezene u krug fabrike sa lokacije $j \in J$ računa se kao proizvod kapaciteta vozila *CV* i ukupnog broja poseta lokaciji $j \in J$ od strane svih vozila;
- Realna promenljiva *T* se koristi za formulisanje funkcije cilja. Označava vremenski trenutak kada sva vozila završe svoju poslednju turu.

3.4. MIQCP formulacija

MIQCP formulacija razmatranog problema raspoređivanja vozila je:

$$\min \quad T \tag{1}$$

pri uslovima

$$\sum_{j\in J} x_{ik}^j \leqslant 1 \quad \forall i \in I, \quad \forall k \in K, \tag{2}$$

$$CV \sum_{(i,k)\in I\times K} x_{ik}^j - CV + \varepsilon \leqslant c_j \quad \forall j \in J, \tag{3}$$

$$T_j + y_j \leqslant 1 \quad \forall j \in J, \tag{4}$$

$$\sum_{j\in J}(1-y_j)\cdot c_j + CV \cdot \sum_{(i,j,k)\in I\times J\times K} y_j x_{ik}^j \geqslant C, \tag{5}$$

$$\sum_{(i,k)\in I\times K} x_{ik}^j \geqslant (c_j/CV)\cdot(1-y_j) \quad \forall j \in J, \tag{6}$$

$$\sum_{(i,k)\in I\times K} y_j x_{ik}^j + \varepsilon \leqslant c_j/CV \quad \forall j \in J, \tag{7}$$

$$t_{i,k+1} \geqslant t_{ik} + \frac{2}{v}\sum_{j\in J} d_j x_{ik}^j + (u+w)\sum_{j\in J} x_{ik}^j \quad \forall i \in I, \quad \forall k \in K \setminus \{k_{max}\}, \tag{8}$$

$$t_{i,k_{max}} + \frac{2}{v}\sum_{j\in J} d_j x_{ik_{max}}^j + (u+w)\sum_{j\in J} x_{ik_{max}}^j \leqslant T \quad \forall i \in I, \tag{9}$$

$$x_{ik}^j \in \{0,1\} \quad \forall i \in I, \quad \forall j \in J, \quad \forall k \in K, \tag{10}$$

$$y_j \in \{0,1\} \quad \forall j \in J, \tag{11}$$

$$t_{ik} \in [t_{start}, t_{end}] \quad \forall i \in I, \quad \forall k \in K. \tag{12}$$

Funkcija cilja (1) zajedno sa ograničenjem (9) minimizuje vremenski trenutak u kojem se završava dnevni transport za sva vozila, odnosno trenutak u kojem poslednje pristiglo vozilo završi istovar robe u svojoj poslednjoj turi.

Ograničenje (2) obezbeđuje da svako vozilo u svakoj turi može da opsluži najviše jednu lokaciju. Dakle, svako vozilo u svakoj turi ima mogućnost da poseti neku lokaciju ili napravi virtualnu turu. Ukupna količina robe koja se preveze sa lokacije $j \in J$ u fabriku ne može biti veća od pripremljene količine c_j na datoj lokaciji, što je obezbeđeno ograničenjem (3). Umanjilac *CV* koji se nalazi na levoj strani ograničenja (3) odražava činjenicu da poslednje vozilo koje posećuje neku lokaciju ne mora biti u potpunosti natovareno. Kako bi količina koja se transportuje sa lokacije $j \in J$ umanjena za veličinu kapacita vozila *CV* bila strogo manja od prikupljene količine c_j na lokaciji $j \in J$, na levoj strani ograničenja (3) je dodata mala pozitivna konstanta ε. To je neophodno u slučaju kada je c_j jednako proizvodu prirodnog broja i kapaciteta vozila *CV* tj. celobrojnom umnošku veličine *CV*. Bez stroge nejednakosti, bilo bi moguće da vozila nepotrebno naparave turu više do lokacije $j \in J$. Na primer, ako je $c_j = 120$t i $CV = 30$t, lokaciju j je potrebno posetiti najviše 4 puta, međutim, bez konstante ε u ograničenju (3), postoji mogućnost da se lokacija j poseti 5 puta, kako je $5 \cdot 30 - 30 \leqslant 120$.

Ograničenje (4) obezbeđuje uslov da se sve količine sa lokacije $j \in J$ moraju prevesti tokom radnog dana, ukoliko na ovoj lokaciji roba stoji na otvorenom duže od t_0 dana. Imajući u vidu definiciju binarnih veličina T_j (videti podsekciju 3.1), u slučaju kada je $t_j \leqslant t_0$, važi $T_j = 0$, a desna strana ograničenja (4) je jednaka 1. Stoga je, u ovom slučaju, ograničenje (4) zadovoljeno nezavisno od transportovane količine sa lokacije j. Sa druge strane, ako je $t_j > t_0$, tada je $T_j = 1$ i lokacija j mora biti ispražnjena tokom radnog dana, što znači da y_j mora imati vrednost 0.

Ograničenje (5) obezbeđuje zadovoljenje dnevnih potreba fabrike. Prva suma na levoj strani relacije (5) odnosi se na količine koje su dovezene sa lokacija koje su ispražnjene, a druga predstavlja količinu robe, dovezene sa neispražnjenih lokacija.

Vrednosti promenljivih y_j su definisane ograničenjima (6) i (7). Ako je lokacija $j \in J$ ispražnjena, y_j uzima vrednost 0 i relacija (6) obezbeđuje da je lokacija j posećena ne manje od c_j/CV puta. U slučaju kada lokacija $j \in J$ nije ispražnjena, tada je $y_j = 1$ i ograničenje (6) je zadovoljeno. Ograničenje (7) obezbeđuje da je svaka neispražnjena lokacija $j \in J$ posećena strogo manje od c_j/CV puta. U slučaju da se radi o neispražnjenoj lokaciji, ograničenje (7) nije aktivno.

Ograničenje (8) odražava činjenicu da svako vozilo $i \in I$ ne može početi novu turu dok se prethodna ne završi. Vreme polaska svake ture ne sme biti manje od vremena polaska prethodne ture, uvećanom za vreme trajanja vožnje u oba smera i zbirom dužine trajanja utovara i istovara. Ako je tura $k \in K \setminus \{k_{max}\}$ vozila $i \in I$ virtualna, obe sume na desnoj strani ograničenja (8) su jednake 0. Na kraju, ograničenja (10)–(12) definišu tip promenljivih koje su korišćene u predloženom modelu.

4. Opšta metoda promenljivih okolina za predloženi VSP

Metoda promenljivih okolina (VNS) je poznata metaheuristika koju su uveli Mladenović i Hansen 1997. godine [15]. Osnovna ideja VNS metode je pretraga prostora rešenja problema korišćenjem različitih struktura okolina. U opštem slučaju, algoritam VNS metode se sastoji iz tri osnovna koraka: faze razmrdavanja (engl. *Shaking Phase*), lokalne pretrage (engl. *Local Search*) i pomeraja (engl. *Move or Not*). U fazi razmrdavanja, bira se rešenje iz okoline tekućeg rešenja, u cilju diversifikacije rešenja, sto pomaže algoritmu da ne završi u lokalnom optimumu. Zadatak lokalne pretrage je pokušaj poboljšanja rešenja iz faze razmrdavanja korišćenjem jedne ili više struktura okolina. Ukoliko lokalna pretraga dostigne rešenje bolje od tekućeg, vrši se zamena tekućeg rešenja novodobijenim, što predstavlja fazu pomeraja. Ova tri koraka se smenjuju do zadovoljenja kriterijuma zaustavljanja. U literaturi postoje brojne varijante VNS metode koje su uspešno primenjene na različite probleme optimizacije: Osnovna VNS metoda (engl. *Basic VNS - BVNS*), Redukovana VNS metoda (engl. *Reduced VNS - RVNS*), Metoda promenljivog spusta (engl. *Variable Neighborhood Descent - VND*), Opšta VNS metoda (engl. *General VNS - GVNS*), Adaptivna VNS metoda (engl. *Skewed VNS - SVNS*), Primalno-dualna VNS metoda (engl. *Primal-dual VNS - PD-VNS*) i druge. Pregled varijanti VNS metode i njihovih primena može se pronaći u [10].

Za rešavanje problema razmatranog u ovom radu predložena je Opšta metoda promenljivih okolina (GVNS). GVNS metoda koristi VND umesto klasične lokalne pretrage, što znači da GVNS pokušava da poboljša rešenje pretražujući okoline na deterministički način [5, 16, 19, 20]. Glavni koraci predložene GVNS za VSP predstavljeni su u Algoritmu 1. Nakon generisanja inicijalnog rešenja, smenjuju se faza razmrdavanja, VND i faza pomeraja sve dok se ne zadovolji kriterijum zaustavljanja. Rešenje dobijeno u fazi razmrdavanja se poboljšava VND pretragom. Ukoliko je poboljšanje bolje od tekućeg rešenja, vrši se zamena tekućeg rešenja novim i pretraga počinje od prve okoline. U suprotnom pretraga nastavlja od naredne okoline u nizu korišćenih struktura okolina. Ovi koraci se smenjuju sve dok se ne zadovolji kriterijum zaustavljanja, koji u predloženoj GVNS implementaciji predstavlja dostignutu gornju granicu vremena izvršavanja (t_{max}). U nastavku ovog odeljka opisani su elementi dizajnirane GVNS metode za opisani VSP.

4.1. Reprezentacija rešenja. Računanje vrednosti funkcije cilja

Rešenje S predloženog problema VSP predstavljeno je u matricom celih nenegativnih brojeva, koja se sastoji iz m vrsta dužine k_{max}. Vrsta i matrice S odgovara vozilu $i \in \{1, \ldots, m\}$ i sadrži cele nenegativne brojeve $s_{ik} \in \{0, 1, \ldots, n\}$, gde s_{ik} označava indeks lokacije koju vozilo i posećuje u turi $k \in \{1, \ldots, k_{max}\}$. Preciznije, ako vozilo i posećuje lokaciju $j \in \{1, \ldots, n\}$ u turi k važi $s_{ik} = j$. U slučaju kada je tura k vozila i virtualna, $s_{ik} = 0$. Zbog jednostavnosti, virtualne ture svakog vozila nalaze se uvek na kraju vrste koja odgovara posmatranom vozilu u matrici S.

Svako generisano rešenje potrebno je evaluirati. Za dopustivo rešenje, vrednosti $s_{ik}, i \in I, k \in K$ određuju raspored tura svih vozila. Na osnovu vrednosti s_{ik}, računaju se vrednosti t_{ik} koje predstavljaju vremena polazaka vozila iz kruga fabrike za svako vozilo $i \in I$ i svaku turu $k \in K$, na sledeći način. Za svaku lokaciju izračuna se vreme opsluživanja te lokacije koje je jednako sumi vremena trajanja putovanja od fabrike do lokacije i nazad i vremena trajanja utovara i istovara. Kako sva vozila započinju svoje prve ture u isto vreme, za svako vozilo $i \in I$, t_{i1} ima vrednost jednaku vremenu početka radnog dana (t_{start}). Vreme polaska t_{id} vozila $i \in I$ u turi

Algoritam 1 GVNS za VSP

Procedure GVNS($ProblemData$,h_{max}, t_{max})
Generate an initial solution S;
repeat
  $h \leftarrow 1$;
  while $h \leqslant h_{max}$ **do**
    $S^{'} \leftarrow$ Shaking (S, h);
    $S^{''} \leftarrow$ Variable Neighborhood Descent($S^{'}$);
    if $S^{''}$ *is better than* S **then**
      $S \leftarrow S^{''}$;
      $h \leftarrow 1$;
    else $h \leftarrow h + 1$;
  end while
until $SessionTime \geqslant t_{max}$

$d \in \{2, \ldots, k_{max}\}$ jednako je vremenu polaska istog vozila u njegovoj prethodnoj turi, uvećanom za vreme potrebno da se opsluži lokacija posećena u prethodnoj turi. Na taj način, obezbeđuje se da vozilo može početi novu turu tek po završetku prethodne. Izračunata vremena polazaka t_{ik} smeštaju se u matricu T_d. Vreme završetka radnog dana za svako vozilo računa se kao vreme povratka u krug fabrike u poslednjoj nevirtualnoj turi, uvećano za neophodno vreme trajanja istovara. Konačno, prema (1) i (9), vrednost funkcije cilja *T* predstavlja maksimum skupa vremena završetaka radnog dana svih vozila.

4.2. Generisanje inicijalnog rešenja

Inicijalno rešenje u predloženoj GVNS metodi generisano je pohlepnom procedurom, koja počinje sortiranjem lokacija na osnovu dva kriterijuma. Najpre se svaka lokacija označava kao hitna ili ne, imajući u vidu broj dana koliko roba stoji na otvorenom na toj lokaciji. Preciznije, ako se roba na lokaciji $j \in J$ nalazi na otvorenom t_j dana i $t_j > t_0$, lokacija j se smatra hitnom. Za hitnu lokaciju j, nije od značaja za koliko dana t_j prelazi gornju granicu od t_0 dana. Hitne lokacije imaju prioritet i moraju se opslužiti tokom radnog dana. U slučaju kada se količine repe prikupljene na hitnim lokacijama ne mogu prevesti raspoloživim skupom vozila u okviru unapred određenog broja tura i pre kraja radnog vremena, smatra se da problem nema dopustivo rešenje. Drugi kriterijum za izbor lokacija koje će biti opslužene je njihova udaljenost od fabrike, u smislu da lokacija $j \in J$ sa manjom udaljenošću od fabrike d_j ima prioritet. Konačno, sortirana lista lokacija ima sledeću strukturu: na vrhu liste nalaze se hitne lokacije sortitane u neopadajući niz prema udaljenosti od fabrike, a za njima dolaze lokacije koje nisu hitne, takođe sortirane u neopadajući niz prema udaljenosti od fabrike.

Nakon opisanog sortiranja lokacija, matrica S inicijalnog rešenja se formira na sledeći način. Uzima se jedna po jedna lokacija iz sortirane liste i njenim indeksom popunjava se prva kolona matrice rešenja. Kada su prve ture za sva vozila formirane, odnosno kada je popunjena prva kolona matrice rešenja, fomiraju se druge ture tj. popunjava se druga kolona matrice, zatim treće itd. Pritom se elementi matrice popunjavaju indeksom jedne lokacije sve dok se posmatrana lokacija ne isprazni, nakon čega se uzima sledeća lokacija sa liste prioriteta. Poslednja nevirtualna tura svakog vozila može početi ukoliko se posećena lokacija može opslužiti u okviru radnog vremena. Ako nije moguće dodeliti lokaciju vozilu tako da ona bude opslužena u okviru radnog vremena, posmatrana tura i sve naredne ture tog vozila postaju virtualne, a proces generisanja rešenja nastavlja se od sledećeg vozila. Kada se potrebe fabrike zadovolje prebroje se ture svakog vozila. Ako je indeks poslednje nevirtualne ture k_i vozila $i \in I$ manji od k_{max}, sve preostale ture $k_i + 1, k_i + 2, \ldots, k_{max}$ postaju virtualne.

Kada je matrica S generisana, računaju se vrednosti matrice T_d. Vremena polazaka svih vozila u prvoj turi jednaka su početku radnog vremena t_{start}, tj. $t_{i1} = t_{start}$, za svako vozilo $i \in I$. Vremena polazaka u preostalim turama, t_{id}, $d = 2, \ldots, k_{max}$ računaju se uvećavanjem vremena polaska istog vozila u prethodnoj turi vremenom koje je neophodno da bi se opslužila posećena lokacija. Na kraju, prebrojavanjem poseta svakoj lokaciji, računaju se vrednosti promenljivih y_j. Ako broj poseta lokaciji j nije manji od c_j/CV, promenljiva y_j ima vrednost 0, u suprotnom je $y_j = 1$.

Lako se može primetiti da je ovako generisano inicijalno rešenje dopustivo. Svaki element s_{ik} matrice S uzima jednu vrednost skupa celih nenegativnih brojeva koja odgovara indeksu lokacije $j \in J$ posećene od strane vozila i u turi k. Preciznije, $s_{ik} = j$ označava da je $x_{ik}^{j} = 1$, sa izuzetkom $s_{ik} = 0$ u slučaju virtualne ture. Dakle, ograničenje (2), koje obezbeđuje uslov da svako vozilo u svakoj turi može posetiti najviše jednu lokaciju ili napraviti praznu turu, je ispunjeno. Kolone matrice inicijalnog rešenja se popunjavaju indeksima lokacija sa liste prioriteta dok se sve hitne lokacije ne isprazne i dok se ne zadovolje potrebe fabrike. Iz tog razloga su ograničenja (4) i (5) takođe zadovoljena. Kako se prilikom popunjavanja matrice rešenja, uzima jedna po jedna lokacija sa liste prioriteta, imajući u vidu količine robe na svakoj od njih, odnosno indeks lokacije se unosi u matricu rešenja sve dok se ta lokacija ne isprazni, ograničenje (3) će biti ispunjeno. Takođe, prilikom popunjavanja matrice rešenja, vozilo započinje nevirtualnu turu jedino u slučaju kada ona može da se završi u okviru radnog vremena, što znači da će sva vozila završiti svoje poslednje ture do t_{end}. Vrednosti promenljivih y_j su definisane tako da zadovoljavaju ograničenja (6) i (7). Vrednosti promenljivih t_{ik}, x_{ik}^j i y_j, dobijene opisanim postupkom očigledno zadovoljavaju ograničenja (8)-(12).

4.3. Strukture okolina

Predložena GVNS metoda koristi četiri strukture okolina, označenih sa: N^I, N^{II}, N^{III} i N^{IV}. Okolina N^I rešenja S definisana je na sledeći način: N^I-sused S' od S se dobija izborom jednog vozila i jedne njegove nevirtualne ture ka lokaciji koja nije hitna, koja se zatim menja nekom drugom neispražnjenom lokacijom. Okolina N^{II} se sastoji iz svih rešenja koja se dobijaju zamenom prve virtualne ture u nizu tura jednog vozila, turom ka nekoj od neispražnjenih lokacija. Okolinu N^{III} resenja S čine sva rešenja koja se dobijaju kada se jedna nevirtualna tura ka nehitnoj lokaciji vozila sa najdužim vremenom završetka radnog dana u rešenju S, zameni virtualnom turom, nakon čega se nova virtualna tura pomera na kraj niza tura posmatranog vozila. Konačno, okolina N^{IV} je definisana na sledeći način: N^{IV}-sused S' od S se dobija tako što vozilo sa najdužim vremenom završetka radnog dana i vozilo sa najkraćim vremenom završetka radnog dana razmenjuju po jednu svoju turu. Treba primetiti da potez kojim je definisana okolina N^{III} može narušiti dopustivost rešenja, jer se može dogoditi da posle poteza u okolini N^{III} potrebe fabrike ne budu zadovoljene. Iz tog razloga, potez u ovoj okolini se prihvata samo ako novo rešenje zadovoljava i ovaj uslov. U suprotnom potez u okolini N^{III} se ponavlja.

4.4. Faza razmrdavanja

Faza razmrdavanja predložene GVNS metode koristi dve strukture okolina: N^I i N^{II}, pri čemu se prvo istražuje okolina N^I za sve vrednosti reda $r = 1, \ldots, r_{max}$, a potom okolina N^{II} samo za $r = 1$. Razmrdavanje u okolini N^I reda r se izvodi ponavljanjem poteza kojim je ta okolina definisana r puta, pri čemu se potezi biraju na slučajan način.

4.5. Faza VND

VND faza koristi strukture okolina N^{III} i N^{IV} na sledeći način. Prvo se deterministički istražuje okolina N^{III}, tako što pokušava da zameni jednu po jednu turu vozila sa najvećim vremenom završetka virtualnom turom. Nakon svakog takvog poteza, proverava se uslov zadovoljenosti potreba fabrike i pretraga u okolini N^{III} posmatranog rešenja se završava najboljim pronađenim susedom u ovoj okolini. Ako je dobijeno poboljšanje bolje od tekućeg VND rešenja, vrši se odgovarajuća zamena i ponavlja se pretraga u okolini N^{III}, u suprotnom se prelazi na pretragu u okolini N^{IV}. Okolina N^{IV} se pretražuje na sledeći način. Pronalaze se dva vozila, m_1 sa najdužim vremenom završetka i m_2 sa nakraćim vremenom završetka radnog dana, i vrši se zamena ture k_1 vozila m_1 ka najudaljenijoj lokaciji u nizu svih njegovih tura sa turom k_2 vozila m_2 ka najbližoj lokaciji od svih lokacija koje vozilo m_2 posećuje. Ukoliko se dobije poboljšanje na nivou dva posmatrana vozila, odnosno ako je maksimum vrednosti njihovih novih vremena završetaka manji od odgovarajuće vrednosti pre izvršene zamene, novo rešenje postaje tekuće VND rešenje i postupak se ponavlja još jednom u okolini N^{IV}. Primetimo da poboljšanje maksimalnog vremena završetaka posmatranog para vozila ne mora obavezno da predstavlja i poboljšanje na nivou celog rešenja. Opisano pretraživanje okoline N^{IV} može dati poboljšanje rešenja u slučaju kada postoje dva vozila sa istom vrednosti vremena završetka koja je pritom i maksimalna u skupu svih vozila. Nakon pretrage okoline N^{IV}, VND algoritam evaluira dobijeno rešenje. Ako je postignuto poboljšanje u odnosu na tekuće VND rešenje, tekuće rešenje se zamenjuje novim i VND pretraga nastavlja istražujući okolinu N^{III}. U suprotnom završava se jedna VND iteracija. Opisane VND iteracije se smenjuju sve dok se dobija poboljšanje tekućeg VND rešenja. Ako su nekoj iteraciji ne ostvari poboljšanje rešenja, VND pretraga se završava, vraćajući najbolje pronađeno rešenje.

5. Eksperimentalni rezultati

Svi eksperimentalni rezultati predstavljeni u ovom odeljku dobijeni su na računaru sa Intel Core i5 - 3320M procesorom, na 2,60 GHz i sa 2 GB RAM memorije pod Windows 7 operativnim sistemom. Eksperimenti su izvršeni nad dva skupa instanci:

- Realne instance dobijene na osnovu podatka iz posmatrane fabrike šećera u Srbiji. Ovaj skup instanci uključuje 43 test primera sa do 15 lokacija, 40 vozila i maksimalno 20 tura tokom radnog dana;

- Generisane instance koje uključuju do 150 lokacija, 100 vozila i maksimalno 100 tura tokom radnog dana. Ovaj skup obuhvata 12 test primera, koji su generisani prateći strukturu realnih instanci.

Za dobijanje optimalnih rešenja problema VSP korišćen je komercijalni rešavač Lingo 17, koji je dizajniran za rešavanje velikog broja različitih tipova matematičkih modela [12]. MIQCP formulacija razmatranog VSP problema je korišćena u okviru Lingo 17 rešavača u cilju dobijanja optimalnih rešenja ili bar gornjih granica. Pritom je postavljena gornja granica vremena rada Lingo rešavača od 5h. Parametar koji se odnosi na kriterijum zaustavljanja u predloženoj GVNS metodi uzima sledeće vrednosti: $t_{max} = 1$s za male realne instance, $t_{max} =$ 10s za realne primere srednjih dimenzija i $t_{max} = 100$s za generisane instance. Ove vrednosti paramatara su izabrane tako da se postigne kompromis između kvaliteta dobijenih rešenja i vremena izvršavanja algoritma. Na svakom razmatranom test primeru, GVNS metoda je izvršavana po 30 puta.

Eksperimentalni rezultati na skupu realnih instanci malih dimenzija za koje je dobijeno optimalno rešenje korišćenjem Lingo rešavača predstavljeni su u Tabeli 1, koja je organizovana na sledeći način. Prva kolona $T_{n,m,k_{max}}$ sadrži naziv instance. U drugoj i trećoj koloni, označenim sa $opt.sol.$ i $t(s)$, predstavljeni su vrednost funkcije cilja koja odgovara optimalnom rešenju dobijenim korišćenjem Lingo rešavača i odgovarajuće vreme rada. Naredne tri kolone se odnose na rezultate testiranja dizajnirane GVNS metode i sadrže: vrednost funkcije cilja najboljeg pronađenog dopustivog rešenja (*best sol.*), pri čemu je korišćena oznaka *opt* u slučaju kada se GVNS rešenje poklapa sa $opt.sol.$, zatim prosečno vreme koje je bilo neophodno GVNS metodi da dostigne svoje najbolje rešenje u 30 uzastopnih izvršavanja i na kraju, prosečno procentualno odstupanje (*avg. gap (%)*) najboljeg GVNS rešenja od optimalnog u 30 uzastopnih izvršavanja.

Na osnovu rezultata predstavljenih u Tabeli 1 može se zaključiti da GVNS za vrlo kratko vreme (u proseku 0,001s) dostiže sva optimalna rešenja koje je pronašao Lingo rešavač za 6,431s u proseku. Može se zaključiti da je GVNS više od 6000 puta brži od Lingo rešavača na skupu realnih test primera malih dimenzija. Osim toga, dizajnirana GVNS metoda je izuzetno stabilna na skupu realnih instanci malih dimenzija, s obzirom da prosečno procentualno odstupanje najboljih GVNS rešenja od optimalnih ima veoma malu vrednost (0,092 %).

Tabela 2 sadrži eksperimentalne rezultate na skupu realnih instanci za koje Lingo rešavač nije uspeo da pronađe optimalno rešenje za 5h, kao i na skupu generisanih instanci većih dimenzija. Prva kolona $T_{n,m,k_{max}}$ sadrži naziv instance. Naredne dve kolone sadrže rezultate testiranja MIQCP formulacije Lingo rešavačem: vrednost funkcije cilja najboljeg pronađenog dopustivog rešenja (*UB*) i odgovarajuću donju granicu vrednosti funkcije cilja (*LB*). Ako u okviru datog vremena izvršavanja nije pronađeno optimalno ili dopustivo rešenje, oznaka $/$ je upisana u odgovarajuće polje. Poslednje tri kolone odnose se na rezultate testiranja predložene GVNS metode i predstavljene su na isti način kao u Tabeli 1.

Levi deo Tabele 2 sadrži rezultate testiranja na skupu realnih instanci srednjih dimenzija, za koje je Lingo rešavač u najboljem slučaju dao dopustivo rešenje za 5h izvršavanja. U slučaju 9 od 12 realnih instanci srednjih dimenzija Lingo rešavac je ponašao dopustiva rešenja koja je GVNS metoda potvrdila ili poboljšala, osim pri testiranju instance $T_{8,6,5}$, a kod koje je GVNS pronašla za nijansu lošije rešenje od Lingo rešavača. Za preostale 3 instance na kojima Lingo rešavač nije uspeo da pronađe čak ni dopustivo rešenje, GVNS je dala rešenja za relativno kratko vreme. Prosečno GVNS vreme izvršavanja na skupu realnih instanci srednjih dimenzija iznosi 0,950s što je više od 18 900 puta brže u poređenju sa vremenom izvršavanja Lingo rešavača (5h). Pored toga, niska vrednost prosečnog procentualnog odstupanja dobijenih GVNS rešenja od najboljih pronađenih u 30 uzastopnih izvršavanja (0,055 %) pokazuje stabilnost predložene GVNS metode na skupu realnih instanci srednjih dimenzija.

Rezultati testiranja generisanih instanci većih dimenzija predstavljeni su u desnom delu Tabele 2. Lingo rešavač nije uspeo da pronađe dopustivo rešenje ni na jednoj generisanoj instanci, dok je GVNS metoda pronašla svoja najbolja rešenja za 24,495s u proseku, sa malim prosečnim procentualnim odstupanjem od najboljih rešenja, dobijenih u 30 uzastopnih puštanja (0,514 %).

Tabela 1. Eksperimentalni rezultati na realnim instancama manjih dimenzija rešenih do optimalnosti

Naziv instance	**Lingo 17**		**GVNS**		
$T_{n,m,k_{max}}$	*opt. sol.*	*t(s)*	*best sol.*	*t(s)*	*avg. gap (%)*
$T_{3,2,4}$	16,874	0,40	*opt*	0,001	0,000
$T_{3,3,2}$	9,780	0,13	*opt*	0,001	0,000
$T_{3,3,3}$	13,727	0,37	*opt*	0,001	0,000
$T_{3,3,4}$	16,303	4,33	*opt*	0,001	0,000
$T_{3,4,2}$	10,580	0,30	*opt*	0,002	0,000
$T_{3,4,3}$	13,727	0,27	*opt*	0,001	0,000
$T_{3,4,4}$	16,303	52,90	*opt*	0,001	0,000
$T_{3,5,2}$	12,294	0,30	*opt*	0,002	0,000
$T_{4,2,3}$	12,870	0,22	*opt*	0,001	0,000
$T_{4,2,4}$	13,446	0,26	*opt*	0,001	0,000
$T_{4,3,2}$	10,580	0,13	*opt*	0,001	0,000
$T_{4,3,3}$	12,299	2,94	*opt*	0,001	0,000
$T_{4,4,2}$	11,151	0,17	*opt*	0,001	0,000
$T_{4,4,3}$	14,299	6,35	*opt*	0,002	0,000
$T_{4,4,4}$	17,446	71,06	*opt*	0,001	0,000
$T_{4,5,2}$	10,866	0,39	*opt*	0,002	2,717
$T_{5,2,3}$	10,584	0,46	*opt*	0,000	0,000
$T_{5,2,4}$	13,156	1,40	*opt*	0,002	0,000
$T_{5,3,2}$	10,580	0,13	*opt*	0,002	0,000
$T_{5,3,3}$	13,156	6,04	*opt*	0,002	0,000
$T_{5,3,4}$	16,589	9,42	*opt*	0,001	0,000
$T_{5,4,3}$	13,156	5,23	*opt*	0,001	0,145
$T_{5,5,2}$	11,437	3,59	*opt*	0,001	0,000
$T_{6,2,3}$	11,727	0,72	*opt*	0,001	0,000
$T_{6,2,4}$	13,160	9,98	*opt*	0,001	0,000
$T_{6,3,2}$	10,294	0,60	*opt*	0,001	0,000
$T_{6,3,3}$	12,870	6,74	*opt*	0,001	0,000
$T_{6,4,2}$	10,009	0,32	*opt*	0,001	0,000
$T_{6,5,2}$	10,866	3,43	*opt*	0,001	0,000
$T_{7,3,3}$	12,299	4,86	*opt*	0,004	0,000
$T_{7,5,2}$	10,580	6,69	*opt*	0,004	0,000
Prosek	12,800	6,431	*opt*	0,001	0,092

Tabela 2. Eksperimentalni rezultati na realnim instancama srednjih dimenzija i generisanim instancama za koje Lingo 17 nije pronašao optimalno rešenje

Naziv instance	**Lingo 17**		**GVNS**			**Naziv instance**	**Lingo 17**		**GVNS**		
$T_{n,m,k_{max}}$	*UB*	*LB*	*best sol.*	*t(s)*	*avg. gap (%)*	$T^r_{n,m,k_{max}}$	*UB*	*LB*	*best sol.*	*t(s)*	*avg. gap (%)*
$T_{4,5,7}$	**15,287**	14,315	**15,287**	0,993	0,000	$T^r_{20,40,20}$	/	/	42,020	25,120	0,136
$T_{5,5,5}$	**18,879**	18,210	**18,879**	0,001	0,000	$T^r_{30,30,25}$	/	/	47,017	12,345	0,454
$T_{5,5,10}$	17,839	16,540	**17,781**	0,003	0,000	$T^r_{30,60,15}$	/	/	29,831	24,745	1,527
$T_{5,10,10}$	17,129	15,974	**17,014**	0,627	0,000	$T^r_{40,60,55}$	/	/	103,966	19,595	0,354
$T_{5,20,20}$	/	/	24,591	2,321	0,205	$T^r_{45,65,70}$	/	/	110,774	1,551	1,001
$T_{6,6,6}$	16,626	16,169	**16,569**	3,355	0,076	$T^r_{50,70,70}$	/	/	117,526	4,260	1,467
$T_{6,7,6}$	11,511	10,765	**11,479**	0,065	0,000	$T^r_{70,60,60}$	/	/	172,653	23,346	0,149
$T_{7,5,6}$	16,426	15,785	**16,140**	0,773	0,110	$T^r_{80,65,65}$	/	/	182,556	45,854	0,272
$T_{8,6,5}$	**14,246**	13,911	14,250	0,097	0,017	$T^r_{90,80,80}$	/	/	159,383	6,956	0,255
$T_{8,40,10}$	/	/	27,920	0,001	0,000	$T^r_{100,85,85}$	/	/	185,36	40,487	0,140
$T_{10,10,10}$	16,924	14,998	**16,243**	0,870	0,000	$T^r_{120,90,90}$	/	/	216,256	31,136	0,010
$T_{15,20,15}$	/	/	26,014	2,298	0,248	$T^r_{150,100,100}$	/	/	240,303	40,148	0,025
Prosek	/	/	18,514	0,950	0,055	**Prosek**	/	/	142,625	24,495	0,514

6. Zaključak

Predmet ovog rada je varijanta problema raspoređivanja vozila koja se odnosi na optimizaciju transporta šećerne repe u Srbiji. Problem je formulisan u vidu Mešovitog celobrojnog programa sa kvadratnim ograničenjima

(MIQCP), koji uključuje sva specifična ograničenja problema i cilj optimizacije. MIQCP model je korišćen u okviru Lingo 17 rešavača, pri čemu su dobijena optimalna rešenja samo za realne test primere malih dimenzija. U cilju rešavanja instanci problema većih dimenzija, dizajnirana je Opšta metoda promenljivih okolina (GVNS) koja je prilagođena razmatranoj varijanti problema raspoređivanja vozila. Predložena GVNS metoda sastoji se iz faze razmrdavanja, metode promenljivog spusta (VND), umesto klasične lokalne pretrage i faze pomeraja. Eksperimentalni rezultati pokazuju da GVNS za vrlo kratko vreme izvršavanja dostiže sva optimalna rešenja dobijena Lingo rešavačem i dostiže ili poboljšava gornje granice na realnim instancama srednjih dimenzija koje nisu rešene do optimalnosti (sa izuzetkom jedne instance). Takođe, dizajnirana GVNS metoda obezbeđuje rešenja na generisanim instancama većih dimenzija za relativno kratko vreme izvršavanja, imajući u vidu dimenzije problema.

Jedan od pravaca budućih istraživanja je kombinovanje dizajnirane GVNS metode sa drugim metodama optimizacije kako bi se poboljšao kvalitet dobijenih rešenja ili dodatno smanjilo vreme izvršavanja, pri rešavanju razmatranog VSP. Takođe, predloženi VSP problem se može proširiti dodavanjem novih ograničenja, na primer, uključivanjem vozila koja nisu homogena ili optimizacijom transporta do dve ili više fabrika za preradu.

Zahvalnica. *Istraživanja na ovom radu su delimično podržana od strane Ministarstva obrazovanja, nauke i tehnološkog razvoja, Republike Srbije, kroz projekte pod brojem OI174010 i OI174033.*

Bibliografija

[1] **D. Agustina, C. Lee, R. Piplani.** Vehicle scheduling and routing at a cross docking center for food supply chains. *International Journal of Production Economics*, 2014, 152, pp. 29-41.

[2] **A. Allahverdi.** The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research*, 2015, 246 (2), pp. 345-378.

[3] **K. Androutsopoulos, K. Zografos.** An integrated modelling approach for the bicriterion vehicle routing and scheduling problem with environmental considerations. *Transportation Research Part C: Emerging Technologies*, 2017, 82, pp. 180-209.

[4] **F. Baita, R. Pesenti, W. Ukovich, D. Favaretto.** A comparison of different solution approaches to the vehicle scheduling problem in a practical case. *Computers & Operations Research*, 2000, 27 (13), pp. 1249-1269.

[5] **J. Brimberg, N. Mladenović, R. Todosijević, D. Urošević.** General variable neighborhood search for the uncapacitated single allocation p-hub center problem. *Optimization Letters*, 2015, 11 (2), pp. 377-388.

[6] **M. Cheikh, M. Ratli, O. Mkaouar, B. Jarboui .** A variable neighborhood search algorithm for the vehicle routing problem with multiple trips. *Electronic Notes in Discrete Mathematics*, 2015, 47, pp. 277-284.

[7] **G.B. Dantzig, D. R. Fulkerson.** Minimizing the number of tankers to meet a fixed schedule. *Naval Research Logistics Quarterly, Wiley Subscription Services, Inc., A Wiley Company*, 1954, 1, pp. 217 - 222.

[8] **J. de Armas, B. Melián-Batista.** Variable neighborhood search for a dynamic rich vehicle routing problem with time windows. *Computers and Industrial Engineering*, 2015, 85, pp. 120-131.

[9] **A. Haghani, M. Banihashemi, K.H. Chiang.** A comparative analysis of bus transit vehicle scheduling models. *Transportation Research Part B: Methodological*, 2003, 37 (4), pp. 301 - 322.

[10] **P. Hansen, N. Mladenovi ć, J.A.M. P érez.** Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 2010, 175 (1), pp. 367-407.

[11] **A. Higgins.** Scheduling of road vehicles in sugarcane transport: A case study at an Australian sugar mill. *European Journal of Operational Research*, 2006, 170 (3), pp. 987 - 1000.

[12] **Lindo Systems Inc.**, `http://www.lindo.com`, 2016.

[13] **R. Macedo, C. Alves, S. Hanafi, B. Jarboui, N. Mladenović, B. Ramos, J.M.V. Carvalho.** Skewed general variable neighborhood search for the location routing scheduling problem. *Computers & Operations Research*, 2015, 61, pp. 143-152.

[14] **E.L. Milan, S.M. Fernandez, L.M.P. Aragones.** Sugar cane transportation in Cuba, a case study. *European Journal of Operational Research*, 2016, 174 (1), pp. 374 - 386.

[15] **N. Mladenovi ć, P. Hansen.** Variable neighborhood search. *Computers & Operations Research*, 1997, 24 (11), pp. 1097 - 1100.

[16] **N. Mladenovi ć, D. Urošević, A. Ilić.** A general variable neighborhood search for the one-commodity pickup-and-delivery travelling salesman problem. *European Journal of Operational Research*, 2012, 220(1), 270âĂŞ285.

[17] **S. Moons, K. Ramaekers, A.A.Y. Caris.** Integrating production scheduling and vehicle routing decisions at the operational decision level: A review and discussion. *Computers and Industrial Engineering*, 2017, 104, pp. 224 - 245.

[18] **S. Thuankaewsing, S. Khamjan, K. Piewthongngam, S. Pathumnakul.** Harvest scheduling algorithm to equalize supplier benefits: A case study from the Thai sugar cane industry. *Computers and Electronics in Agriculture*, 2015, 110, pp. 42 - 55.

[19] **R. Todosijević, A. Mjirda, M. Mladenović, S. Hanafi, B. Gendron.** A general variable neighborhood search variants for the travelling salesman problem with draft limits. *Optimization Letters*, 2017, 11 (6), pp. 1047 - 1056.

[20] **R. Todosijević, D. Urošević, N. Mladenović, S. Hanafi.** A general variable neighborhood search for solving the uncapacitated r-allocation p-hub median problem. *Optimization Letters*, 2017, 11 (6), pp. 1109 âĂŞ 1121.

[21] **H. Wang, J. Shen.** Heuristic approaches for solving transit vehicle scheduling problem with route and fueling time constraints. *Applied Mathematics and Computation*, 2007, 190 (2), pp. 1237 - 1249.

[22] **Y. Xiao, A. Konak.** A simulating annealing algorithm to solve the green vehicle routing & scheduling problem with hierarchical objectives and weighted tardiness. *Applied Soft Computing*, 2015, 34, pp. 372 - 388.