Transportni problem

Motivacija i postavka problema

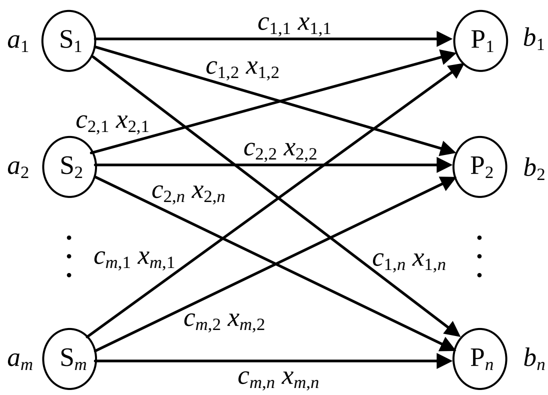
Izučavanje ***problema transporta*** primjenom *analitičkih metoda* potiče iz polovine prošlog stoljeća. Tim se problemom bavilo više poznatih istraživača, među kojima su značajni rezultati ruskog naučnika *L. V. Kantoroviča* iz 1939. godine i američkog naučnika *F. L. Hitchcocka*-a iz 1941. godine. Nekako u isto vrijeme, veliki istraživački napori i rezultati su bili vezani za zadatak *linearnog programiranja*. Tom prilikom se pokazalo da su transportni problemi zapravo *specijalni slučajevi problema linearnog programiranja*, tako da oni u načelu *mogu biti riješeni ma kojim od metoda za rješavanje zadataka linearnog programiranja* (npr. simpleks metodom). Međutim, isto tako se pokazalo da, zbog svoje *specifične strukture*, problemi transporta mogu biti riješeni i *mnogo efikasnijim metodima* u odnosu na opće metode linearnog programiranja. Tako je *Dantzig* uskoro nakon otkrića simpleks metoda razvio i specijalan metod za rješavanje transportnog problema zasnovan na *simpleks množiteljima*. *Vogel* je predložio *aproksimativni* metod za nalaženje početnog rješenja, koji tipično daje rješenje, koje ako nije optimalno, nije ni daleko od optimalnog i moguće ga je relativno brzo unaprijediti do optimalnog pomoću drugih metoda. Prvi dokazano efikasan metod za pronalaženje optimalnog rješenja transportnih problema predložili su 1953. godine *Charnes* i *Cooper*. Njihov metod poznat je danas pod imenom "metod skakanja s kamena na kamen" (engl. *Stepping-Stone Method*) koji je kasnije, razvojem *teorije dualnosti*, unaprijeđen do tzv. *MODI metoda*. Konačno, treba istaći i da su *Ford* i *Fulkerson* 1956. godine objavili još jedan metod za rješavanje transportnog problema, koji je zapravo *u tijesnoj vezi* sa njihovim poznatim metodom za nalaženje *maksimalnog protoka u transportnim mrežama*.

Naziv "transportni problem" potiče od činjenice da su su se problemima ove vrste isprva modelirali i rješavali *problemi transporta* koji su se sastojali u nalaženju *optimalnog plana transporta* koji će obezbijediti *minimalne troškove transporta* kroz *zadanu transportnu mrežu* i uz *zadana transportna sredstva*. Međutim, danas se u probleme ovog tipa uključuju i mnogo raznovrsniji zadaci, poput problema *optimalnog razmještaja mašina*, *postrojenja*, *službi*, *skladišta*, *servisa*, *energetskih objekata*, *prodajnih objekata*, itd.

Da bismo uvidjeli kako izgleda *matematski model transportnih problema*, krenućemo od klasičnog problema nalaženja *optimalnog plana transporta*. U ovakvim problemima najčešće se javlja zadatak *transporta jedne vrste robe* iz određenog broja *mjesta skladištenja* (kratko *skladišta*) u određeni broj *mjesta potrošnje* (kratko *potrošača*), a da se pri tome ostvare minimalni troškovi transporta. Pri tome se smatra da su poznati sljedeći parametri:

- Broj *ishodišta* (*skladišta*) m;
- Broj *odredišta* (*potrošača*) n;
- *Količine robe* a_i, $i = 1 .. m$ koje su smještene u skladištima S_i, $i = 1 .. m$;
- *Količine robe* b_j, $j = 1 .. n$ koje potražuju potrošači P_j, $j = 1 .. n$;
- *Troškovi transporta* $c_{i,j}$, $i = 1 .. m$, $j = 1 .. n$ po *jednoj količinskoj jedinici robe* koji nastaju pri transportu robe iz skladišta S_i, $i = 1 .. m$ ka potrošaču P_j, $j = 1 .. n$.

Ono što je potrebno odrediti su nepoznate veličine $x_{i,j}$, $i = 1 .. m$, $j = 1 .. n$ koje predstavljaju *količine robe* koje treba transportovati iz skladišta S_i, $i = 1 .. m$ u odredište P_j, $j = 1 .. n$. Šematski se problem transporta može prikazati sljedećom slikom:



U praktično svim transportnim problemima, *raspoložive količine* a_i, $i = 1 \,..\, m$ i *potraživane količine* b_j, $j = 1 \,..\, n$ uvijek moraju biti *striktno pozitivne veličine*, tako da će se u nastavku smatrati da su ovi parametri pozitivni brojevi. Što se tiče *troškova* $c_{i,j}$, $i = 1 \,..\, m$, $j = 1 \,..\, n$, oni su uglavnom *nenegativni* (nula je također smislena vrijednost, jer je $c_{i,j} = 0$ kad god *nema troškova transporta* iz skladišta S_i u odredište P_j, što se dešava recimo ukoliko se skladište S_i i odredište P_j nalaze *na istom mjestu*), mada je ponekad *moguće i da budu negativni*. Naime, negativna vrijednost $c_{i,j}$ znači da se umjesto *troškova* odgovarajućim transportom *ostvaruje neka dobit*. Što se tiče nepoznatih veličina $x_{i,j}$, $i = 1 \,..\, m$, $j = 1 \,..\, n$, one uvijek *moraju biti nenegativne*, jer transport *negativnih količina robe nema nikakvog smisla*.

Izvedimo sada matematski model transportnog problema. Prvo ćemo razmotriti tzv. ***balansirani slučaj***. Troškovi koji nastaju pri transportu $x_{i,j}$ količinskih jedinica robe po jediničnoj cijeni transporta $c_{i,j}$ iznose $c_{i,j}\, x_{i,j}$, $i = 1 \,..\, m$, $j = 1 \,..\, n$, tako da ukupna cijena trasnporta iznosi

$$\begin{aligned} Z = {} & c_{1,1}\, x_{1,1} + c_{1,2}\, x_{1,2} + \ldots + c_{1,n}\, x_{1,n} + \\ & + c_{2,1}\, x_{2,1} + c_{2,2}\, x_{2,2} + \ldots + c_{2,n}\, x_{2,n} + \\ & \ldots \\ & + c_{m,1}\, x_{m,1} + c_{m,2}\, x_{m,2} + \ldots + c_{m,n}\, x_{m,n} \end{aligned}$$

Očigledno je da ukupna količina robe koja se transportira iz i-tog skladišta ka svim potrošačima iznosi $x_{i,1} + x_{i,2} + \ldots + x_{i,n}$ i ona mora biti jednaka količini robe a_i koja se nalazi u tom skladištu, što nameće ograničenje oblika

$$x_{i,1} + x_{i,2} + \ldots + x_{i,n} = a_i$$

Ovakva ograničenja se mogu napisati za sva skladišta S_i, $i = 1 \,..\, m$. Slično, ukupna količina robe koja se transportira ka j-tom potrošaču iz svih skladišta iznosi $x_{1,j} + x_{2,j} + \ldots + x_{m,j}$ i ona mora biti jednaka količini b_j koju potražuje taj potrošač, što nameće ograničenje oblika

$$x_{1,j} + x_{2,j} + \ldots + x_{m,j} = b_j$$

Ovakva ograničenja se mogu napisati za sve potrošače P_j, $i = 1 \,..\, n$. Zajedno sa ograničenjima na nenegativnost promjenljivih, ovo daje sljedeći matematski model transportnog problema:

$$\begin{aligned} \arg\min Z = {} & c_{1,1}\, x_{1,1} + c_{1,2}\, x_{1,2} + \ldots + c_{1,n}\, x_{1,n} + \\ & + c_{2,1}\, x_{2,1} + c_{2,2}\, x_{2,2} + \ldots + c_{2,n}\, x_{2,n} + \\ & \ldots \\ & + c_{m,1}\, x_{m,1} + c_{m,2}\, x_{m,2} + \ldots + c_{m,n}\, x_{m,n} \end{aligned}$$

p.o.

$$\begin{aligned} & x_{1,1} + x_{1,2} + \ldots + x_{1,n} = a_1 \\ & x_{2,1} + x_{2,2} + \ldots + x_{2,n} = a_2 \\ & \ldots \\ & x_{m,1} + x_{m,2} + \ldots + x_{m,n} = a_m \\ & x_{1,1} + x_{2,1} + \ldots + x_{m,1} = b_1 \\ & x_{1,2} + x_{2,2} + \ldots + x_{m,2} = b_2 \\ & \ldots \\ & x_{1,n} + x_{2,n} + \ldots + x_{m,n} = b_n \\ & x_{i,j} \ge 0,\ i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n \end{aligned}$$

Uz oznake za sumaciju, ovaj model se može kompaktnije zapisati na sljedeći način:

$$\arg\min Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{i,j}\ x_{i,j}$$

p.o.

$$\sum_{j=1}^{n} x_{i,j} = a_i,\ \ i = 1 \,..\, m$$

$$\sum_{i=1}^{m} x_{i,j} = b_j,\ \ j = 1 \,..\, n$$

$$x_{i,j} \ge 0,\ \ i = 1 \,..\, m,\ \ j = 1 \,..\, n$$

Vidimo da se u ovom problemu javljaju *tri skupine ograničenja*: ograničenja na *isporuke iz ishodišta*, ograničenja na *potrebe odredišta* i ograničenja na *nenegativnost promjenljivih*.

Kako je funkcija cilja linearna, i kako su sva ograničenja linearna, vidimo da transportni problem spada u probleme linearnog programiranja (činjenica da su promjenljive iz praktičnih razloga *indeksirane sa dva indeksa* ne mijenja ovu činjenicu). Ukupan broj promjenljivih koji se javlja u ovom modelu je $m\,n$, dok je broj ograničenja (od kojih su sva tipa jednakosti) $m + n$. Dakle, matrica ograničenja ovog problema, posmatranog kao problem linearnog programiranja, ima dimenzije $(m + n) \times m\,n$. Međutim, zbog činjenice da je ova matrica *vrlo specijalnog oblika*, transportni problemi se vrlo efikasno rješavaju usprkos tipično *veoma velikom broju promjenljivih* koje se javljaju u ovim problemima. Recimo, svi koeficijenti matrice ograničenja su *isključivo nule ili jedinice* i u svakoj koloni takve matrice javljaju se *tačno dvije jedinice*. Ovaj specijalni oblik matrice ograničenja omogućava razvoj veoma efikasnih algoritama za rješavanje ovakvih problema.

Transportni problemi, u obliku kako su prikazani gore, *rješivi su samo ukoliko kapaciteti izvorišta i odredišta zadovoljavaju tzv.* ***uvjet balansiranosti***

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

Ovaj uvjet zapravo znači da *ukupna količina robe u svim skladištima* mora biti jednaka *ukupnoj potražnji svih potrošača* (odnosno, *ukupna ponuda mora biti jednaka ukupnoj potražnji*). Neophodnost ovog uvjeta vrlo je jednostavno pokazati. Naime, saberemo li *sva ograničenja na isporuke iz ishodišta*, dobijamo jednakost

$$\sum_{i=1}^{m} \sum_{j=1}^{n} x_{i,j} = \sum_{i=1}^{m} a_i$$

S druge strane, saberemo li *sva ograničenja na potrebe odredišta*, dobijamo jednakost

$$\sum_{j=1}^{n} \sum_{i=1}^{m} x_{i,j} = \sum_{j=1}^{n} b_j$$

Lijeve strane ove i prethodne nejednakosti su praktično identične (razlikuju se *samo u poretku sumiranja*), tako da *moraju biti identične i njihove desne strane*, što nije ništa drugo nego uvjet balansiranosti. Ukoliko ovaj uvjet nije zadovoljen, *ograničenja problema nisu međusobno saglasna*, tako da problem, u formi kakva je postavljena ovdje, *nema rješenja* (uskoro ćemo uopćiti model transportnog problema da omogući i rješivost slučajeva u kojima uvjet balansiranosti *nije ispunjen*).

Iz prethodnog izvođenja slijedi još jedna interesantna stvar. Naime, zaključili smo da je *suma svih ograničenja na isporuke iz ishodišta jednaka sumi svih ograničenja na potrebe odredišta*. Iz ovoga slijedi da sva ograničenja *nisu međusobno nezavisna*, nego se svako od ograničenja *može izraziti preko onih preostalih* (kao suma svih ograničenja iz suprotne skupine u odnosu na skupinu u kojoj se razmatrano ograničenje nalazi minus suma preostalih ograničenja iz one skupine u kojoj se razmatrano ograničenje nalazi). Drugim riječima, jedno od ograničenja (bilo koje) se *može slobodno odbaciti*, a da pri tome *model ostane i dalje ispravan*. Kao posljedica ovoga, transportni problem efektivno ima samo $m + n - 1$ a ne $m + n$ ograničenja. Slijedi da *ma kakvo bazno* (*a samim tim i optimalno*) *rješenje transportnog problema* može imati *najviše* $m + n - 1$ *nenultih promjenljivih*. U tom smislu je svako bazno rješenje transportnog problema uvijek *barem jednostruko degenerirano*. Međutim, ta barem jednostruka degeneriranost se kod transportnih problema *prećutno podrazumijeva*, odnosno smatra se da je bazno rješenje degenerirano samo ako ono sadrži *manje od* $m + n - 1$ *nenultih promjenljivih*.

Transportni problemi kod kojih je zadovoljen uvjet balansiranosti nazivaju se ***balansirani*** odnosno ***zatvoreni transportni problemi***. Ukoliko uvjet balansiranosti nije ispunjen, *matematski model transportnog problema mora se drugačije formirati*, jer smo vidjeli da gore formiran model *uopće nije rješiv* ukoliko uvjet balansiranosti nije ispunjen. Transportni problemi kod kojih uvjet balansiranosti *nije zadovoljen* nazivaju se ***nebalansirani*** odnosno ***otvoreni transportni problemi***.

Razmotrimo prvo slučaj u kojem je *ukupna ponuda veća od ukupne potražnje*, odnosno slučaj u kojem je umjesto uvjeta balansiranosti zadovoljen uvjet

$$\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$$

U ovom slučaju je jasno da će *barem u nekom od skladišta* S_i, $i = 1 \,..\, m$ ostati izvjesna količina robe *koja neće biti isporučena nigdje*. Slijedi da ukupna isporuka iz i-tog skladišta koja iznosi $x_{i,1} + x_{i,2} + \ldots + x_{i,n}$ ne mora nužno biti jednaka količini robe a_i koja se nalazi u tom skladištu, nego može biti *i manja od toga*. Zbog toga se ograničenja na isporuke iz ishodišta mijenjaju tako da će njihov tip umjesto ograničenja tipa jednakosti postati ograničenja tipa "manje ili jednako". Drugim riječima, matematski model problema postaje

$$\arg\min Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{i,j}\, x_{i,j}$$

p.o.

$$\sum_{j=1}^{n} x_{i,j} \le a_i,\ \ i = 1 \,..\, m$$

$$\sum_{i=1}^{m} x_{i,j} = b_j,\ \ j = 1 \,..\, n$$

$$x_{i,j} \ge 0,\ \ i = 1 \,..\, m,\ \ j = 1 \,..\, n$$

Pokažimo kako se ovaj model može svesti na model *balansiranog transportnog problema*. Ukoliko u prvu skupinu ograničenja uvedemo *dopunske* (*izravnavajuće*) *promjenljive* koje možemo označiti sa $x_{i,\,n+1}$, $i = 1 \,..\, m$, model dobija oblik

$$\arg\min Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{i,j}\, x_{i,j}$$

p.o.

$$\sum_{j=1}^{n} x_{i,j} + x_{i,\,n+1} = a_i,\ \ i = 1 \,..\, m$$

$$\sum_{i=1}^{m} x_{i,j} = b_j,\ \ j = 1 \,..\, n$$

$$x_{i,j} \ge 0,\ \ i = 1 \,..\, m,\ \ j = 1 \,..\, n$$

Saberemo li sada sva ograničenja iz prve skupine i sva ograničenja iz druge skupine, dobićemo sljedeće jednakosti:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{i,j} + \sum_{i=1}^{m} x_{i,n+1} = \sum_{i=1}^{m} a_i$$

$$\sum_{j=1}^{n}\sum_{i=1}^{m} x_{i,j} = \sum_{j=1}^{n} b_j$$

Kako je prva dvojna suma sa lijeve strane prve jednakosti jednaka dvojnoj sumi sa lijeve strane druge jednakosti, slijedi

$$\sum_{i=1}^{m} x_{i,n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$$

Označimo li veličinu sa desne strane (koja je pozitivna) sa b_{n+1} i pridružimo ovu jednakost drugoj skupini ograničenja, te definiramo da je $c_{i,\,n+1} = 0$, $i = 1 \,..\, m$, model dobija oblik

$$\arg\min Z = \sum_{i=1}^{m} \sum_{j=1}^{n+1} c_{i,j}\, x_{i,j}$$

p.o.

$$\sum_{j=1}^{n+1} x_{i,j} = a_i,\; i = 1\,..\,m$$

$$\sum_{i=1}^{m} x_{i,j} = b_j,\; j = 1\,..\,n+1$$

$$x_{i,j} \geq 0,\; i = 1\,..\,m,\; j = 1\,..\,n$$

Novodobijeni model je očito model *balansiranog transportnog problema*, kod kojeg se umjesto n potrošača javlja $n + 1$ potrošač (da je uvjet balansiranosti ovdje ispunjen, jasno je iz načina kako je definirana veličina b_{n+1}). Intuitivno, ovo znači da je problem sveden na balansirani problem uvođenjem jednog dodatnog ***fiktivnog potrošača*** P_{n+1} koji na sebe "preuzima" višak ponude u odnosu na potražnju. Njegove "potrebe" su upravo jednake iznosu tog "viška". U stvarnosti, roba koja se "isporučuje" fiktivnom potrošaču zapravo *ostaje u skladištu*, te je jasno da su cijene transporta iz ma kojeg skladišta ka fiktivnom potrošaču jednake nuli.

Na sličan način se nebalansirani transportni problemi kod kojih je *ukupna potražnja veća od ukupne ponude* svode na balansirane transportne probleme uvođenjem jednog dodatnog ***fiktivnog skladišta*** S_{m+1} koje "snabdjeva" potrošače neophodnim količinama koje nedostaju u stvarnim skladištima. Kapacitet ovog fiktivnog skladišta jednak je iznosu "manjka" ponude u odnosu na potražnju. U stvarnosti, roba koja se "nabavlja" iz fiktivnog skladišta zapravo *neće biti niodakle nabavljena*, nego će se morati naknadno nabaviti na *neki drugi način* ili iz *nekih drugih izvora* (koji nisu uključeni u postavku problema). Jasno je da su cijene transporta iz ovog fiktivnog skladišta ka ma kojem potrošaču jednake nuli, jer se taj transport u stvarnosti i ne vrši.

Mada su transportni problemi obično problemi *minimizacije*, mogu se razmatrati i varijante transportnih problema kod kojih je cilj *maksimizacija*. Na primjer, ukoliko se transport ne posmatra sa aspekta *vlasnika robe* (koji mora *platiti* za transport i želi proći *što jeftinije*), nego sa aspekta *prevoznika* (koji *zarađuje* vršeći transport i želi *zaraditi što više*), transportni problem postaje problem maksimizacije. Problemi maksimizacije se vrlo jednostavno svode na probleme minimizacije *množenjem funkcije cilja sa* -1, kao uostalom u svim problemima matematičkog programiranja.

U transportnim problemima se često dešava i da *transport iz nekog specifičnog skladišta* S_p ka *nekom specifičnom potrošaču* P_q iz nekog razloga *nije dozvoljen* (npr. potrošač P_q iz nekog razloga jednostavno *ne želi* robu iz skladišta S_p), odnosno kod kojih mora biti $x_{p,q} = 0$. Ovakve situacije se vrlo jednostavno svode na klasični transportni problem *upotrebom kaznenih koeficijenata*, odnosno postavljanjem da je $c_{p,q} = M$ gdje je M neki *vrlo veliki broj*. Na ovaj način, transportu iz skladišta S_p ka potrošaču P_q daje se *izuzetno nepovoljna cijena*, koja će onemogućiti da se ovaj transport nađe u optimalnom rješenju. Mada bi teoretski trebalo uzeti $M = \infty$, u praksi je dovoljno da M bude 2–3 reda veličine veći od cijena ostalih transporta.

S obzirom da smo pokazali da se nebalansirani transportni problemi *mogu uvijek svesti na ekvivalentne balansirane probleme*, u nastavku ćemo, bez umanjenja općenitosti, razmatrati *isključivo balansirane transportne probleme*. Iz izlaganja koja će biti provedena u nastavku, biće jasno da je za transportne probleme izražene u balansiranom obliku uvjet balansiranosti *ne samo neophodan, nego i dovoljan* za postojanje optimalnog rješenja. S obzirom da se nebalansirani problemi uvijek mogu svesti na balansirane probleme (kod kojih je uvjet balansiranosti automatski ispunjen), slijedi da su *svi transportni problemi uvijek rješivi*, odnosno da uvijek *posjeduju optimalno rješenje*.

Već je rečeno da se, u načelu, transportni problemi *mogu rješavati simpleks algoritmom*. Međutim, zbog velikog broja promjenljivih koje se tipično javljaju u transportnim problemima, takvo rješavanje je *prilično neefikasno*. Srećom, koristeći *specifičnu strukturu transportnih problema*, za njihovo rješavanje moguće je razviti *mnogo efikasnije algoritme*. Pored toga, *transportni problemi posjeduju još neke lijepe osobine koje opći problemi linearnog programiranja ne posjeduju*. Recimo, iz izlaganja koja će uslijediti biće jasno da *ukoliko su svi kapaciteti* a_i, $i = 1\,..\,m$ i b_j, $j = 1\,..\,n$ *cijeli brojevi*, tada je *svako bazno rješenje transportnog problema također cjelobrojno* (što naravno uključuje i optimalno rješenje). Drugim riječima, *eventualni zahtjevi na cjelobrojnost promjenljivih nimalo ne otežavaju rješavanje transportnih problema* (pod uvjetom da su svi zadani kapaciteti također cjelobrojni, što praktično uvijek jesu kad god se zahtijevaju cjelobrojna rješenja), odnosno *takvi zahtjevi će biti automatski ispunjeni*.

Određivanje početnog dopustivog baznog rješenja

Većina do sada razvijenih metoda za rješavanje transportnih problema pretpostavlja da je prvo na neki način *već određeno jedno početno dopustivo bazno rješenje*, koje se onda koristi kao *polazna osnova* za nalaženje optimalnog rješenja (mada postoje i neki metodi, kao što je *Ford-Fulkersonov metod* ili *metod uvjetno optimalnih planova*, koji *ne zahtijevaju nalaženje početnog dopustivog baznog rješenja*). Zbog toga je razvijen veći broj *pomoćnih metoda* za nalaženje *jednog dopustivog baznog rješenja*. Najpoznatiji takvi metodi su sljedeći:

- *Metod sjeverozapadnog ugla*;
- *Metod minimalnih jediničnih troškova* (ili *metod najmanjeg elementa u matrici cijena*);
- *Vogelov aproksimativni metod*.

Prvo ćemo objasniti ***metod sjeverozapadnog ugla***, koji je *konceptualno najjednostavniji*, ali i *najlošiji*. Ukoliko nepoznate promjenljive $x_{i,j}$, $i = 1\,..\,m$, $j = 1\,..\,n$ posmatramo kao *matricu*, tada dodjeljivanje početnih vrijednosti promjenljivim počinje od *gornjeg lijevog ugla matrice* (što zapravo odgovara *sjeverozapadu* na geografskim kartama, odakle potiče ime metoda), i to na sljedeći način:

- Promjenljiva $x_{1,1}$ dobija *maksimalnu moguću vrijednost* koja je *nije veća ni od kapaciteta* a_1 *ni od kapaciteta* b_1 (tj. koja je *ograničena odozgo manjim od ova dva kapaciteta*), odnosno uzima se $x_{1,1} = \min\{a_1, b_1\}$.
- Prethodna dodjela će ili u potpunosti iscrpiti skladište S_1 (ako je $a_1 < b_1$), ili će u potpunosti podmiriti potrošača P_1 (ako je $a_1 > b_1$), ili oboje (ako je $a_1 = b_1$). Ako je skladište u potpunosti iscrpljeno, eliminiramo ga iz daljeg razmatranja (odnosno eliminiramo iz razmatranja odgovarajući red matrice, tj. prvi red), a kapacitet (potražnju) potrošača umanjujemo za dodijeljeni iznos $x_{1,1}$. Ako je potrošač u potpunosti podmiren, eliminiramo ga iz daljeg razmatranja (odnosno eliminiramo iz razmatranja odgovarajuću kolonu matrice), a kapacitet skladišta umanjujemo za dodijeljeni iznos $x_{1,1}$.

Postupak se dalje nastavlja sa promjenljivom koja se nalazi u gornjem lijevom uglu nakon obavljene eliminacije jednog skladišta ili potrošača (ili i jednog i drugog). To će biti promjenljiva $x_{2,1}$ ako je eliminirano skladište S_1, promjenljiva $x_{1,2}$ ako je eliminiran potrošač P_1, odnosno $x_{2,2}$ ako je eliminirano oboje. Dalji postupak se vrši na analogan način *pražnjenjem skladišta* odnosno *zadovoljavanjem potražnji* sve dok se *ne rasporede sve zalihe u skladištima* odnosno dok se *ne zadovolji sva tražnja*. Sve promjenljive koje *nisu učestvovale u postupku* imaće *vrijednost nula*.

Iz samog toka postupka, jasno je da ovaj postupak *zaista daje jedno dozvoljeno rješenje*, za koje nije teško pokazati ni da je *bazno rješenje*. S obzirom da je ovaj postupak *uvijek moguće provesti*, jasno je da transportni problemi *uvijek imaju dozvoljenih rješenja*.

➢ ***Primjer***: U nekom transportnom problemu sa tri skladišta S_1, S_2 i S_3 i četiri potrošača P_1, P_2, P_3 i P_4, kapaciteti skladišta (zalihe), potrebe potrošača i cijene transporta po količinskoj jedinici robe dati su u sljedećoj tabeli:

	P_1	P_2	P_3	P_4	Zalihe
S_1	8	9	4	6	100
S_2	6	9	5	3	120
S_3	5	6	7	4	140
Potrebe	90	125	80	65	

Odrediti jedno dopustivo rješenje ovog problema koristeći metod sjeverozapadnog ugla.

Postupak započinjemo dodjelom $x_{1,1} = \min\{100, 90\} = 90$. Ovim je potrošač P_1 potpuno podmiren, pa ga isključujemo iz razmatranja, dok u skladištu S_1 ostaje količina od $100 - 90 = 10$ jedinica. Nova situacija prikazana je u sljedećoj tabeli. Dodijeljena vrijednost prikazana je u *gornjoj lijevoj polovini* odgovarajućeg polja tabele, kako se obično radi, dok je kolona koja se isključuje iz razmatranja *osjenčena* (alternativno se, umjesto sjenčenja, koristi i *precrtavanje*):

	P_1	P_2	P_3	P_4	Zalihe
S_1	90 8	9	4	6	10
S_2	6	9	5	3	120
S_3	5	6	7	4	140
Potrebe	0	125	80	65	

Gornji lijevi (sjeverozapadni) ugao preostalog dijela tabele sada odgovara promjenljivoj $x_{1,2}$, pa vršimo dodjelu $x_{1,2} = \min\{10, 125\} = 10$. Ovim iscrpljujemo skladište S_1 te ga isključujemo iz razmatranja, dok potražnju potrošača P_2 smanjujemo za dodijeljeni iznos 10, tj. na iznos $125 - 10 = 115$. Nova situacija prikazana je u sljedećoj tabeli:

	P_1	P_2	P_3	P_4	Zalihe
S_1	90 8	10 9	4	6	0
S_2	6	9	5	3	120
S_3	5	6	7	4	140
Potrebe	0	115	80	65	

Nastavljamo sada sa promjenljivom $x_{2,2}$, kojoj dodjeljujemo iznos $x_{2,2} = \min\{120, 115\} = 115$. Ovim se potpuno podmiruje potrošač P_2, kojeg isključujemo iz razmatranja, dok u skladištu S_2 ostaje količina od $120 - 115 = 5$ jedinica. Nova situacija prikazana je u sljedećoj tabeli.

	P_1	P_2	P_3	P_4	Zalihe
S_1	90 8	10 9	4	6	0
S_2	6	115 9	5	3	5
S_3	5	6	7	4	140
Potrebe	0	0	80	65	

Sada je na redu promjenljiva $x_{2,3}$, kojoj dodjeljujemo vrijednost $x_{2,3} = \min\{5, 80\} = 5$. Na ovaj način se iscrpljuje skladište S_2 te ga isključujemo iz razmatranja, dok potražnju potrošača P_3 smanjujemo na iznos $80 - 5 = 75$. Ovim se dobija situacija kao u sljedećoj tabeli:

	P_1	P_2	P_3	P_4	Zalihe
S_1	90 8	10 9	4	6	0
S_2	6	115 9	5 5	3	0
S_3	5	6	7	4	140
Potrebe	0	0	75	65	

Sljedeća je na redu promjenljiva $x_{3,3}$, kojoj dodjeljujemo iznos $x_{3,3} = \min\{140, 75\} = 75$. Ovim se potpuno podmiruje potrošač P_3, kojeg isključujemo iz razmatranja, dok u skladištu S_2 ostaje količina od $140 - 75 = 65$ jedinica. Nova situacija prikazana je u sljedećoj tabeli.

	P_1	P_2	P_3	P_4	Zalihe
S_1	90 8	10 9	4	6	0
S_2	6	115 9	5 5	3	0
S_3	5	6	75 7	4	65
Potrebe	0	0	0	65	

Konačno, posljednja je na redu promjenljiva $x_{3,4}$, kojoj dodjeljujemo iznos $x_{3,4} = \min\{65, 65\} = 65$. Nakon ove dodjele, iscrpljava se skladište S_3, ali se također u potpunosti podmiruje potrošač P_4, te oboje isključujemo iz razmatranja. Ovim su *raspodijeljene sve zalihe* i *zadovoljene sve potrebe*, čime je nađeno jedno *dozvoljeno bazno rješenje*. Ova situacija je prikazana u sljedećoj tabeli u kojoj *nisu prikazani kapaciteti skladišta niti potrebe potrošača*, jer će u nastavku biti pokazano da ove vrijednosti *više nizašta ne trebaju nakon što se nađe jedno dopustivo bazno rješenje*. Drugim riječima, ove vrijednosti se koriste samo u početnim predradnjama za pronalaženje početnog dopustivog baznog rješenja, dok u kasnijem toku postupka za pronalaženje optimalnog rješenja, *one više nisu potrebne*.

	P_1	P_2	P_3	P_4
S_1	90 / 8	10 / 9	4	6
S_2	6	115 / 9	5 / 5	3
S_3	5	6	75 / 7	65 / 4

Promjenljive kojima nije dodijeljena nikakva vrijednost, u ovom početnom rješenju imaju vrijednost 0. Dakle, početno rješenje dobijeno na ovaj način glasi:

$x_{1,1} = 90$ $x_{1,2} = 10$ $x_{1,3} = 0$ $x_{1,4} = 0$
$x_{2,1} = 0$ $x_{2,2} = 115$ $x_{2,3} = 5$ $x_{1,4} = 0$
$x_{3,1} = 0$ $x_{3,2} = 0$ $x_{3,3} = 75$ $x_{3,4} = 65$

Mada je metod sjeverozapadnog ugla *veoma jednostavan*, njegov nedostatak je što on gotovo uvijek daje početno dopustivo rešenje koje je *veoma daleko od optimalnog*. Na primjer, troškovi transporta koji slijede iz ovog rješenja iznose:

$$Z = c_{1,1}\,x_{1,1} + c_{1,2}\,x_{1,2} + c_{2,2}\,x_{2,2} + c_{2,3}\,x_{2,3} + c_{3,3}\,x_{3,3} + c_{3,4}\,x_{2,3} =$$
$$= 8\cdot 90 + 9\cdot 10 + 9\cdot 115 + 5\cdot 5 + 7\cdot 75 + 4\cdot 65 = 2655$$

Međutim, ovakvo rješenje je veoma daleko od optimalnog (naime, pokazaćemo kasnije da optimalni troškovi transporta u ovom primjeru iznose 1830). Stoga je, u kasnijem toku postupka, potreban priličan broj iteracija da se ovo rješenje unaprijedi do optimalnog rješenja. Na primjer, vidjećemo da je potrebno čitavih 5 iteracija nekih od klasičnih metoda za popravku rješenja da se ovo rješenje unaprijedi do optimalnog rješenja, bez obzira što se radi o veoma malom problemu. Zbog toga su razvijeni alternativni metodi za nalaženje početnog dopustivog rješenja koje tipično daju početno rješenje koje je znatno bliže optimalnom rješenju u odnosu na rješenje koje se dobija metodom sjeverozapadnog ugla.

Nije teško shvatiti zbog čega metod sjeverozapadnog ugla tipično daje posve loše početno rješenje. Naime, ovaj metod u *potpunosti ignorira cijene transporta* $c_{i,j}$, nego mu je jedini cilj *da obezbijedi zadovoljenje ograničenja*. Prvi pokušaj da se uzmu u obzir i *cijene transporta* dovodi do metoda poznatog kao ***metod minimalnih jediničnih troškova*** (odnosno ***metod najmanjeg elementa u matrici cijena***). Ovaj metod je *gotovo isto tako jednostavan* kao i metod sjeverozapadnog ugla, a *obično daje mnogo bolje početno bazno rješenje*. To je klasični primjer *pohlepnog metoda*, koji u svakom koraku pokušava da obavi *što je god moguće veći transport preko polja sa minimalnim troškovima* u onim redovima odnosno kolonama gdje ponuda odnosno potražnja nisu iscrpljene (za razliku od metoda sjeverozapadnog ugla koji *uvijek cilja na polje u gornjem lijevom uglu neovisno od cijene transporta*). U ostalim aspektima, metod minimalnih jediničnih troškova izvodi se na identičan način kao i metod sjeverozapadnog ugla.

Demonstrirajmo ovaj metod na istom primjeru na kojem je bio demonstriran i metod sjeverozapadnog ugla. Polje sa najmanjim troškovima transporta je $c_{2,4} = 3$, pa ćemo prvo pokušati što je god moguće veći transport obaviti preko ovog polja. Kako potrebe potrošača P_4 iznose 65, dok skladište S_2 ima zalihe 120, maksimalni mogući transport $S_2 – P_4$ iznosi $x_{2,4} = \min\{120, 65\} = 65$. Ovim se potpuno podmiruje potrošač P_4, kojeg eliminiramo iz daljeg razmatranja (i prikazujemo ga osjenčeno ili ga precrtavamo) dok se zalihe skladišta S_2 smanjuju na iznos 120 – 65 = 55. Time se dobija situacija kao u sljedećoj tabeli:

	P_1	P_2	P_3	P_4	Zalihe
S_1	8	9	4	6	100
S_2	6	9	5	65 / 3	55
S_3	5	6	7	4	140
Potrebe	90	125	80	0	

Sada, u nepodmirenom (neosjenčenom) dijelu tabele minimalne troškove transporta ima polje $c_{1,3} = 4$, pa ćemo izvršiti maksimalno mogući transport $x_{1,3} = \min\{100, 80\} = 80$, preko ovog polja. Na taj način se potpuno podmiruje potrošač P_3, kojeg eliminiramo iz daljih razmatranja, a nove zalihe skladišta S_1 iznose $100 - 80 = 20$. Ovo je prikazano u sljedećoj tabeli:

	P_1	P_2	P_3	P_4	Zalihe
S_1	8	9	80 / 4	6	20
S_2	6	9	5	65 / 3	55
S_3	5	6	7	4	140
Potrebe	90	125	0	0	

Minimalne troškove u nepodmirenom dijelu tabele sada ima ima polje $c_{3,1} = 5$, pa vršimo maksimalno mogući transport $x_{3,1} = \min\{140, 90\} = 90$ preko ovog polja, čime se potpuno podmiruje potrošač P_1, a nove zalihe skladišta S_3 iznose $140 - 90 = 50$. Ovo je prikazano u sljedećoj tabeli:

	P_1	P_2	P_3	P_4	Zalihe
S_1	8	9	80 / 4	6	20
S_2	6	9	5	65 / 3	55
S_3	90 / 5	6	7	4	50
Potrebe	0	125	0	0	

Sada je najmanje polje u preostalom dijelu tabele $c_{3,2} = 6$, pa ćemo obaviti maksimalno mogući transport $x_{3,2} = \min\{50, 125\} = 50$ preko ovog polja. Ovim se potpuno iscrpljuje skladište S_3 dok nove potrebe potrošača P_2 iznose $125 - 50 = 75$. Ovim se dobija situacija kao u sljedećoj tabeli:

	P_1	P_2	P_3	P_4	Zalihe
S_1	8	9	80 / 4	6	20
S_2	6	9	5	65 / 3	55
S_3	90 / 5	50 / 6	7	4	0
Potrebe	0	75	0	0	

U sljedećem koraku, preostala su samo dva polja u neprecrtanom dijelu tabele, koja imaju istu cijenu transporta 9. Nasumično ćemo se odlučiti za jedno od ova dva polja, recimo za polje $c_{1,2} = 9$ i obavićemo maksimalno mogući transport $x_{1,2} = \min\{20, 75\} = 20$ preko ovog polja. Ovim se potpuno iscrpljuje skladište S_1, dok nove potrebe potrošača P_2 iznose $75 - 20 = 55$. Novonastala situacija prikazana je u sljedećoj tabeli:

	P_1	P_2	P_3	P_4	Zalihe
S_1	8	20 / 9	80 / 4	6	0
S_2	6	9	5	65 / 3	55
S_3	90 / 5	50 / 6	7	4	0
Potrebe	0	55	0	0	

Konačno, u posljednjem koraku obavljamo transport $x_{2,2} = 55$ preko jedinog preostalog polja, čime se u potpunosti iscrpljuju sva skladišta i podmiruju svi potrošači. Ovim dobijamo početno dopustivo rješenje prikazano u sljedećoj tabeli:

	P_1	P_2	P_3	P_4
S_1	8	20 / 9	80 / 4	6
S_2	6	55 / 9	5	65 / 3
S_3	90 / 5	50 / 6	7	4

Drugim riječima, dobijamo početno bazno rješenje u kojem je

$$x_{1,2} = 20 \qquad x_{1,3} = 80 \qquad x_{2,2} = 55 \qquad x_{2,4} = 65 \qquad x_{3,1} = 90 \qquad x_{3,2} = 50$$

dok su sve ostale vrijednosti $x_{i,\,j}$ nule. Troškovi transporta za ovakvo rješenje iznose

$$Z = 9 \cdot 20 + 4 \cdot 80 + 9 \cdot 55 + 3 \cdot 65 + 5 \cdot 90 + 6 \cdot 50 = 1940$$

Kao što vidimo, dobijeno polazno dopustivo rješenje je znatno bolje nego rješenje dobijeno metodom sjeverozapadnog ugla (mada ni ono nije optimalno).

Još bolje početno rješenje može se dobiti ukoliko se napadne na *slabe tačke* metoda minimalnih jediničnih troškova. Osnovna mana ovog metoda je njegova *pohlepa*, koja se očitava u tome da se u *svakom koraku po svaku cijenu* pokušava ostvariti *maksimalan transport* preko polja koje *u tom trenutku* (među raspoloživim poljima) ima *najpovoljniju jediničnu cijenu transporta*. Međutim, takva strategija može nas dovesti do toga da u *kasnijim koracima moramo birati posve nepovoljne varijante*, jer smo *prebrzo iscrpili povoljne varijante*. Stoga je razvijen jedan heuristički metod nazvan ***Vogelov aproksimativni metod*** (poznat i pod skraćenim nazivom ***VAM***) koji uklanja ovaj nedostatak. Ovaj metod se zasniva na *izbjegavanju velikog mogućeg žaljenja*. Naime, pretpostavimo da zbog nekog razloga *nismo u mogućnosti da transport iz nekog snabdjevača odnosno u neki potrošač obavimo preko najpovoljnijeg polja* u nekom redu odnosno koloni, nego smo *prisiljeni da obavimo transport preko sljedećeg po povoljnosti polja* u istom redu odnosno koloni. Razliku u cijeni najpovoljnijeg i sljedećeg po povoljnosti (dakle, razliku između dva najmanja elementa) u nekom redu odnosno koloni zvaćemo ***moguće žaljenje*** ili ***kajanje*** (engl. *regret*) za taj red i kolonu, s obzirom da će se *za toliko povećati jedinična cijena transporta ukoliko zbog nekog razloga propustimo najpovoljniji transport u tom redu odnosno koloni*. Vogelov aproksimativni metod u svakom koraku pokušava da obavi najveći mogući transport preko polja sa minimalnim jediničnim troškovima transporta, ali *samo u onim redovima odnosno kolonama koje imaju najveću vrijednost mogućeg žaljenja*. U svemu ostalom, *ovaj metod je identičan metodi minimalnih jediničnih troškova*.

Primijenimo sada Vogelov aproksimativni metod na isti problem kao u prethodnom primjeru. Za tu svrhu, početnu tabelu proširićemo jednim novim redom i jednom novom kolonom koju možemo nazvati “Žaljenje” (tačnije bi bilo “Moguće žaljenje”). U ovaj red odnosno kolonu upisuju se *razlike između dva najmanja elementa* za svaku kolonu odnosno za svaki red. Recimo, prvi element u novododanoj koloni je 2, jer su dva najmanja elementa u odgovarajućem redu 4 i 6, a njihova razlika iznosi 2. Na taj način, dobijamo sljedeću početnu tabelu:

	P_1	P_2	P_3	P_4	Zalihe	Žaljenje
S_1	8	9	4	6	100	2
S_2	6	9	5	3	120	2
S_3	5	6	7	4	140	1
Potrebe	90	125	80	65		
Žaljenje	1	3	1	1		

Vidimo da najveće moguće žaljenje (3) imamo u koloni P_2. To znači da ukoliko zbog nekog razloga propustimo da obavimo transport preko najpovoljnijeg polja u toj koloni $c_{3,2} = 6$, prva sljedeća "prilika" po povoljnosti u toj koloni biće polje sa cijenom transporta 9, tako da "moguće žaljenje" iznosi 3 jedinice. Za sve ostale redove i kolone to "moguće žaljenje" je manje. Odlučićemo se stoga da maksimalno mogući transport obavimo upravo preko najpovoljnijeg polja u toj koloni. Taj transport iznosi $x_{3,2} = \min\{140, 125\} = 125$, čime se u potpunosti podmiruje potrošač P_2, a zalihe skladišta S_3 smanjuju se na $140 - 125 = 15$. Nakon što eliminiramo kolonu P_2, potrebno je ponovo izračunati "moguća žaljenja" za sve redove, koja se mogu promijeniti jer je kolona P_2 izbačena "iz igre". To se u ovom slučaju nije desilo, tako da dobijamo situaciju prikazanu u sljedećoj tabeli:

	P_1	P_2	P_3	P_4	Zalihe	Žaljenje
S_1	8	9	4	6	100	2
S_2	6	9	5	3	120	2
S_3	5	125 / 6	7	4	15	1
Potrebe	90	0	80	65		
Žaljenje	1	3	1	1		

Sada, najveće moguće žaljenje imamo u redovima S_1 i S_2. Polje sa najpovoljnijom cijenom u tim redovima je $c_{2,4} = 3$, tako da obavljamo najveći mogući transport $x_{2,4} = \min\{120, 65\} = 65$. Ovim se u potpunosti podmiruje potrošač P_4, dok se zalihe skladišta S_2 smanjuju na $120 - 65 = 55$. Nakon eliminiranja kolone P_2 promijeniće se moguća žaljenja za sve redove (na primjer, za prvi red dva najmanja elementa nakon što su kolone P_2 i P_4 ispale "iz igre" iznose 4 i 8, tako da je moguće žaljenje 4). Stoga se dobija situacija prikazana u sljedećoj tabeli:

	P_1	P_2	P_3	P_4	Zalihe	Žaljenje
S_1	8	9	4	6	100	4
S_2	6	9	5	65 / 3	55	1
S_3	5	125 / 6	7	4	15	2
Potrebe	90	0	80	0		
Žaljenje	1	3	1	1		

Najveće moguće žaljenje sada je u redu S_1, tako da prebacujemo najveći mogući transport $x_{1,3} = \min\{100, 80\} = 80$ preko najpovoljnijeg polja $c_{1,3} = 4$ u tom redu, a zalihe skladišta S_1 smanjujemo na $100 - 80 = 20$. Nakon eliminiranja reda P_2, ostao je samo jedan red (P_1), pa se sva "moguća žaljenja" za redove svode na nulu, jer nemamo dva različita najmanja elementa (drugim riječima, nemamo šta "žaliti" u tim redovima, jer *nemamo ni alternative*). U suštini, nakon što ostane samo jedan neprecrtani red ili kolona, postupak se praktički *nastavlja identično kao kod metode minimalnih jediničnih troškova*, tako da u principu redove i kolone za "žaljenje" više i ne treba voditi. Ipak, radi potpunosti, ovdje će i oni biti prikazani, tako da sada imamo situaciju kao u sljedećoj tabeli:

	P_1	P_2	P_3	P_4	Zalihe	Žaljenje
S_1	8	9	80 / 4	6	20	0
S_2	6	9	5	65 / 3	55	0
S_3	5	125 / 6	7	4	15	0
Potrebe	90	0	0	0		
Žaljenje	1	3	1	1		

Sada je jasno da je najpovoljnije obaviti transport $x_{3,1} = \min\{15, 90\} = 15$ preko polja $c_{3,1} = 5$, čime se u potpunosti iscrpljuje skladište S_3 dok se potrebe potrošača P_1 smanjuju na $90-15 = 75$, čime se dobija situacija prikazana u sljedećoj tabeli:

	P_1	P_2	P_3	P_4	Zalihe	Žaljenje
S_1	8	9	80 / 4	6	20	0
S_2	6	9	5	65 / 3	55	0
S_3	15 / 5	125 / 6	7	4	0	0
Potrebe	75	0	0	0		
Žaljenje	2	3	1	1		

U sljedećem koraku obavljamo transport $x_{2,1} = \min\{55, 75\} = 55$ preko polja $c_{2,1} = 5$, čime se u potpunosti iscrpljuje skladište S_2 dok se potrebe potrošača P_1 smanjuju na $75-55 = 20$. Novonastala situacija prikazana je u sljedećoj tabeli:

	P_1	P_2	P_3	P_4	Zalihe	Žaljenje
S_1	8	9	80 / 4	6	20	0
S_2	55 / 6	9	5	65 / 3	0	0
S_3	15 / 5	125 / 6	7	4	0	0
Potrebe	20	0	0	0		
Žaljenje	0	3	1	1		

Konačno, u posljednjem koraku obavljamo transport $x_{1,1} = 20$ preko jedinog preostalog polja, čime se u potpunosti iscrpljuju svi snabdjevači i podmiruju svi potrošači. Ovim dobijamo početno dopustivo bazno rješenje prikazano u sljedećoj tabeli:

	P_1	P_2	P_3	P_4
S_1	20 / 8	9	80 / 4	6
S_2	55 / 6	9	5	65 / 3
S_3	15 / 5	125 / 6	7	4

Drugim riječima, dobijamo rješenje

$x_{1,2} = 20$ $\quad x_{1,3} = 80$ $\quad x_{2,1} = 55$ $\quad x_{2,4} = 65$ $\quad x_{3,1} = 15$ $\quad x_{3,2} = 125$

pri čemu su ostale vrijednosti $x_{i,j}$ nule. Troškovi transporta za ovakvo rješenje iznose

$$Z = 8 \cdot 20 + 4 \cdot 80 + 6 \cdot 55 + 3 \cdot 65 + 5 \cdot 15 + 6 \cdot 125 = 1830$$

Ovdje dobijamo vrlo interesantnu situaciju da je dobijeno početno dopustivo rješenje *ujedno i optimalno*. Naravno, ovo ne mora uvijek da se desi. Međutim, Vogelov aproksimativni metod, čak i kada ne da odmah optimalno rješenje, daje početno rješenje koje *tipično nije daleko od optimalnog*. S obzirom da je Vogelov aproksimativni metod heurističke prirode, teško je *egzaktno procijeniti kvalitet metoda*. Međutim, statistička testiranja ovog metoda na velikom broju praktičnih primjera pokazala su da ovaj metod u oko 70 % manjih praktičnih primjera *odmah dovede do početnog dopustivog rješenja koje je ujedno i optimalno*, dok je u svega oko 30 % slučajeva potrebno njegovo dalje popravljanje.

Unapređenje tekućeg rješenja do dostizanja optimuma

Kao što je već ranije najavljeno, većina metoda za nalaženje optimalnog rješenja transportnog problema polazi od *ma kojeg dopustivog baznog rješenja*, koje se onda unapređuje kroz *konačan niz iteracija* sve dok se ne *dostigne optimalno rješenje*. Početno dopustivo bazno rješenje je načelno moguće odrediti na *bilo koji način*, recimo *bilo kojim od metoda opisanih u prethodnom odjeljku*, mada se preporučuje da se koristi *Vogelov aproksimativni metod*, jer on *tipično daje početno rješenje najbliže optimalnom*, pa je *tipično potrebno manje iteracija da se dostigne optimum* nego ako se krene od nekog drugog početnog rješenja (mada ni ovo nije uvijek pravilo).

Od poznatih metoda za unapređenje rješenja transportnog problema najpoznatiji je metod poznat pod nazivom ***metod skakanja s kamena na kamen*** odnosno (po izvornom engleskom nazivu) ***stepping-stone metod***, te njegova znatno unaprijeđena modifikacija poznata kao ***MODI*** metod. Oba pomenuta metoda zasnovana su na *istoj ideji*, a razlikuju se samo u tome kako se određuju neke veličine na osnovu kojih se donose izvjesne odluke u ovim metodima. U osnovi, i jedan i drugi metod su zapravo *prerušeni simpleks metod*, samo posebno prilagođen specifičnoj formi koju imaju transportni problemi.

Osnovna ideja ovih metoda je u tom da se u tabeli koja odgovara *tekućem dopustivom baznom rješenju* ispituju *nezauzeta polja*, odnosno polja u koja nije upisana *nikakva vrijednost transporta*. Ta polja zapravo odgovaraju *nebaznim promjenljivim* koje u tekućem baznom rješenju imaju vrijednost nula, odnosno $x_{i,j} = 0$. To znači da se, u takvom rješenju, između skladišta S_i i potrošača P_j *ne vrši nikakav transport* (u žargonu ćemo reći da se *ne vrši transport preko odgovarajućeg polja u tabeli*). Cilj ispitivanja je da se za svako takvo polje ustanovi da li bi se *uključenjem transporta preko tog polja* (tj. transporta između skladišta i potrošača koji odgovaju tom polju) ukupni trošak *povećao ili smanjio, ili bi možda ostao isti* (jedina razlika između stepping-stone i *MODI* metoda je u tome *kako se vrši ovo ispitivanje*). Kada se nađe polje čije uključenje u transport *smanjuje ukupni trošak* (ovome u simpleks metodu odgovara *izbor promjenljive koja ulazi u bazu*), vrši se *preraspodjela transporta* vodeći pri tome računa da se *ne naruši dopustivost rješenja* (ovome u simpleks metodu odgovara *transformacija simpleks tabele pri prelasku sa jednog na drugo dozvoljeno bazno rješenje*). Pri tome će zahtjev da novo rješenje i dalje bude bazno i dopustivo dovesti do toga da će *neko polje ostati bez transporta*, odnosno da će odgovarajuća promjenljiva *postati jednaka nuli* (ovome u simpleks metodu odgovara *izbor promjenljive koja napušta bazu*). Optimalno rješenje se dostiže onog trenutka kada *nije moguće naći niti jednu preraspodjelu transporta koja bi smanjila ukupan trošak transporta*.

Mada je *MODI* metod praktičniji za rad, ovdje će prvo, kao konceptualno jednostavniji, biti opisan ***metod skakanja s kamena na kamen*** odnosno ***stepping-stone metod***, koji se u literaturi može naći i pod brojnim drugim imenima, kao što su ***metod raspodjele***, ***distributivni metod***, ***metod šahovske kule***, itd. Ovaj metod započinje izračunavanjem tzv. ***relativnih koeficijenata troškova*** $d_{i,j}$ za svako polje za koje je $x_{i,j} = 0$ koji pokazuju *za koliko bi se promijenili ukupni troškovi transporta ukoliko bismo jednu količinsku jedinicu transporta preraspodijelili da se vrši preko ovog polja* (zapravo, i *MODI* metod započinje sa izračunavanjem ovih koeficijenata, samo što je *način njihovog izračunavanja* kod *MODI* metoda potpuno drugačiji). Metod stepping-stone ove koeficijente računa na intuitivno vrlo jednostavan način. Jasno je da ako želimo preraspodijeliti izvjesnu količinu transporta preko nekog polja u i-tom redu i j-toj koloni, tada *za isti iznos moramo umanjiti neki transport u i-tom redu i u j-toj koloni* da bi ograničenja bila zadovoljena. Međutim, to umanjenje transporta će opet zahtijevati da se za isti iznos povećaju transporti u nekim poljima u koloni odnosno redu koji odgovaraju poljima u kojima je umanjen transport, itd. Očigledno će ravnoteža biti očuvana samo *ukoliko se uspije uspostaviti zatvoreni ciklus* koji počinje od polja koje testiramo, a koji *prolazi kroz polja sa nenultim transportom*, u kojem će se naizmjenično povećavati i smanjivati transport za isti iznos.

Može se pokazati da pod uvjetom da tekuće rješenje *nije degenerirano* (slučaj degeneracije ćemo razmotriti nešto kasnije), odnosno da postoji tačno $m+n-1$ polja sa nenultim transportom, takav ciklus *uvijek postoji* i *jedinstven je*. Konkretno, takav ciklus uvijek ima oblik *pravougaonog poligona* (tj. poligona čije su sve stranice horizontalne ili vertikalne), čiji jedan vrh leži u *polju koje se testira*, a ostali vrhovi u *nekim poljima sa nenultim transportom*. Takav poligon može imati *najmanje* 4, a *najviše* $m+n$ vrhova. Pri ručnom radu, u problemima manjih dimenzija takav poligon se obično lako uočava prostim posmatranjem tablice transporta. Nešto kasnije ćemo razmotriti i *algoritamski pristup* nalaženju takvog poligona, odnosno ciklusa sa traženim svojstvom.

Kada smo za neko polje formirali odgovarajući poligon – ciklus, odgovarajući relativni koeficijent troškova $d_{i,j}$ dobijamo *sabiranjem* koeficijenata $c_{i,j}$ u onim poljima duž ciklusa u kojima će doći do *povećanja* transporta a *oduzimanjem* koeficijenata $c_{i,j}$ u onim poljima duž ciklusa u kojima će doći do *smanjenja* transporta. Drugim riječima, $d_{i,j}$ se dobija *alternativnim sabiranjem i oduzimanjem cijena* $c_{i,j}$ duž odgovarajućeg ciklusa.

Nakon što se odrede koeficijenti $d_{i,j}$ za sva polja sa nultim trenutnim transportom, određuje se *postoji li preraspodjela transporta koja bi mogla umanjiti ukupne troškove* (na ovom mjestu se metod stepping-stone i *MODI* metod ponovo sastaju i nastavljaju istim tokom). Ukoliko su svi koeficijenti $d_{i,j}$ *pozitivni*, svaka moguća preraspodjela transporta na odgovarajuće polje *samo bi uvećala troškove transporta*, što zapravo znači da je tekuće rješenje *ujedno i optimalno*. Ukoliko među koeficijentima $d_{i,j}$ ima i *negativnih*, tada bi preraspodjela transporta na odgovarajuće polje mogla *umanjiti ukupni trošak*, i rješenje je potrebno unaprijediti. Ukoliko su svi koeficijenti $d_{i,j}$ *nenegativni*, ali među njima ima i koeficijenata koji su *jednaki nuli*, rješenje *nije moguće unaprijediti*, te je ono *također optimalno*, ali bi preraspodjela transporta preko ma kojeg polja za koje je $d_{i,j} = 0$ ostavila *troškove transporta nepromijenjenim*. Drugim riječima, u ovom slučaju, *optimalno rješenje nije jedinstveno*.

Iz izloženog slijedi da preraspodjelu *moramo vršiti* kad god je makar jedan od koeficijenata $d_{i,j}$ negativan. Ako je takvih koeficijenata više, preraspodjelu možemo u principu vršiti na *ma koje polje sa negativnim relativnim koeficijentom troškova*, ali se obično bira ono polje u kojem je koeficijent $d_{i,j}$ *najnegativniji* (ovo odgovara Dantzigovom pravilu pivotiranja kod simpleks algoritma). Da bismo ponovo dobili bazno rješenje, preraspodjelu moramo vršiti tako da se transport "praznog polja" uveća *što je god više moguće*, a da se pri tom *ne naruši dopustivost rješenja*. To zapravo znači da je maksimalna količina koju smijemo preraspodijeliti jednaka *najmanjem od svih transporta koji ćemo pri tome umanjiti*. Nakon obavljene preraspodjele, upravo će polje koje sadrži taj najmanji transport *ostati u potpunosti bez transporta*, odnosno ono će postati novo "prazno polje". Postupak se sada ponavlja ispočetka, sve dok se ne dostigne rješenje koje nije moguće više unaprijediti.

- ***Primjer*** : Koristeći metod stepping-stone, pronaći optimalno rješenje transportnog problema postavljenog u primjeru iz prethodng odjeljka polazeći od početnog dopustivog rješenja koje je određeno metodom sjeverozapadnog ugla.

U prethodnom odjeljku smo, primjenom metoda sjeverozapadnog ugla, došli do početnog dopustivog rješenja koje je predstavljeno sljedećom tabelom:

	P_1	P_2	P_3	P_4
S_1	90 / 8	10 / 9	4	6
S_2	6	115 / 9	5 / 5	3
S_3	5	6	75 / 7	65 / 4

Pri tome je vrijednost funkcije cilja koja odgovara ovom rješenju bila $Z = 2655$. Bitno je uočiti i da ovo rješenje *nije degenerirano*, jer je broj polja sa nenultim transportom jednak 6, što je tačno jednako vrijednosti $m+n-1 = 3+4-1 = 6$.

Kako su *prazna polja* odnosno polja bez dodijeljenog nenultog transporta polja koja odgovaraju promjenljivim $x_{1,3}$, $x_{1,4}$, $x_{2,1}$, $x_{2,4}$, $x_{3,1}$ i $x_{3,,2}$ (kraće ćemo umjesto *polje koje odgovara promjenljivoj* $x_{1,3}$ govoriti samo *polje* $x_{1,3}$ itd.), potrebno je odrediti odgovarajuće relativne koeficijente troškova $d_{1,3}$, $d_{1,4}$, $d_{2,1}$, $d_{2,4}$, $d_{3,1}$ i $d_{3,,2}$. Odredimo prvo koeficijent $d_{1,3}$. Da bismo *povećali transport* preko polja $x_{1,3}$, neophodno je *umanjiti transport* bilo preko polja $x_{1,1}$ bilo preko polja $x_{1,2}$ da bi ograničenje za skladište S_1 ostalo na snazi. Međutim, umanji li se transport preko polja $x_{1,1}$, *nećemo imati gdje da povećamo transport da bi ograničenje za potrošača* P_1 *ostalo na snazi*. Stoga *jedino možemo umanjiti transport preko polja* $x_{1,2}$. To će opet zahtijevati da se poveća transport preko polja $x_{2,2}$ (da se očuva ograničenje za potrošača P_2) što dalje zahtijeva smanjenje transporta preko polja $x_{2,3}$ (da se očuva ograničenje za skladište S_2). Ovo se savršeno uklapa sa početnim povećanjem transporta preko polja $x_{1,3}$. Dakle, uspjeli smo formirati smislen ciklus povećavanja i smanjenja transporta koja čuva valjanost svih ograničenja i taj ciklus konkretno glasi $x_{1,3} \rightarrow x_{1,2} \rightarrow x_{2,2} \rightarrow x_{2,3} \rightarrow x_{1,3}$. Pri tome se transport povećava u poljima $x_{1,3}$ i $x_{2,2}$, a smanjuje u poljima $x_{1,2}$ i $x_{2,3}$, tako da je

$$d_{1,3} = c_{1,3} - c_{1,2} + c_{2,2} - c_{2,3} = 4 - 9 + 9 - 5 = -1$$

Nađimo sada koeficijent $d_{1,4}$. Povećamo li transport preko polja $x_{1,4}$, ponovo je neophodno umanjiti transport preko polja $x_{1,1}$ ili preko polja $x_{1,2}$. Na isti način kao pri određivanju koeficijenta $d_{1,3}$ zaključujemo da umanjivanje transporta preko polja $x_{1,1}$ ne dolazi u obzir, nego se umanjuje transport preko polja $x_{1,2}$. Dalje, ovo zahtijeva povećanje trasnporta preko polja $x_{2,2}$, smanjenje transporta preko polja $x_{2,3}$, zatim povećanje transporta preko polja $x_{3,3}$ te, konačno, smanjenje transporta preko polja $x_{3,4}$, čime se ciklus zatvara. Traženi ciklus glasi $x_{1,4} \rightarrow x_{1,2} \rightarrow x_{2,2} \rightarrow x_{2,3} \rightarrow x_{3,3} \rightarrow x_{3,4} \rightarrow x_{1,4}$, tako da koeficijent $d_{1,4}$ iznosi:

$$d_{1,4} = c_{1,4} - c_{1,2} + c_{2,2} - c_{2,3} + c_{3,3} - c_{3,4} = 6 - 9 + 9 - 5 + 7 - 4 = 4$$

Preostale koeficijente $d_{2,1}$, $d_{2,4}$, $d_{3,1}$ i $d_{3,,2}$ nalazimo na analogan način, i oni iznose:

$$d_{2,1} = c_{2,1} - c_{2,2} + c_{1,2} - c_{1,1} = 6 - 9 + 9 - 8 = -2$$

$$d_{2,4} = c_{2,4} - c_{2,3} + c_{3,3} - c_{3,4} = 3 - 5 + 7 - 4 = 1$$

$$d_{3,1} = c_{3,1} - c_{3,3} + c_{2,3} - c_{2,2} + c_{1,2} - c_{1,1} = 5 - 7 + 5 - 9 + 9 - 8 = -5$$

$$d_{3,2} = c_{3,2} - c_{3,3} + c_{3,2} - c_{2,2} = 6 - 7 + 5 - 9 = -5$$

Kako je već ranije opisano, relativni koeficijenti troškova predstavljaju *promjenu ukupnih troškova* ako bi se izvršila *preraspodjela transporta* po navedenim poljima za *jednu jedinicu transportovane robe*. U prosjeku, *najveće umanjenje troškova* se može očekivati ako izvršimo preraspodjelu transporta na polje kojem odgovara *najnenegativniji* koeficijent (mada ovo ne mora biti uvijek tačno, jer umanjenje troškova ovisi i od količine koju smijemo preraspodijeliti). U ovom slučaju, postoje dva najnegativnija koeficijenta $d_{3,1}$ i $d_{3,2}$, od kojih ćemo se nasumično odlučiti recimo za $d_{3,1}$. Stoga ćemo *uvećati transport* preko polja $x_{3,1}$ (tj. iz skladišta S_1 ka potrošaču P_3) *što je god moguće više*, a da se pri tome *ne naruši dopustivost rješenja* (tj. da sva ograničenja za skladišta i potrošače ostanu zadovoljena i da sve promjenljive ostanu nenegativne). Kako ciklus koji smo imali pri određivanju $d_{3,1}$ glasi $x_{3,1} \rightarrow x_{3,3} \rightarrow x_{2,3} \rightarrow x_{2,2} \rightarrow x_{1,2} \rightarrow x_{1,1} \rightarrow x_{3,1}$, pokušaj preraspodjele transporta t količinskih jedinica preko polja $x_{3,1}$ slikovito je predstavljen sljedećom shemom:

	P_1	P_2	P_3	P_4
S_1	$90-t$ / 8	$10+t$ / 9	4	6
S_2	6	$115-t$ / 9	$5+t$ / 5	3
S_3	t / 5	6	$75-t$ / 7	65 / 4

Povećamo li transport preko polja $x_{3,1}$ za t količinskih jedinica, odmah za isti iznos moramo smanjiti transport preko polja $x_{3,3}$, povećati transport preko polja $x_{2,3}$, smanjiti transport preko polja $x_{2,2}$, povećati transport preko polja $x_{1,2}$ te konačno smanjiti transport preko polja $x_{1,1}$, čime ograničenja za skladišta i potrošače ostaju zadovoljena. Drugim riječima, *uvećanje transporta* u jednom polju uvijek je praćeno *umanjenjem transporta* u poljima koja su s njim u istom redu i istoj koloni, a pripadaju odgovarajućem ciklusu. U ovom primjeru, uvećavaju se transporti $x_{3,1}$, $x_{2,3}$ i $x_{1,2}$, a umanjuju transporti $x_{3,3}$, $x_{2,2}$ i $x_{1,1}$. Da bi nakon preraspodjele transporta rješenje ostalo dopušteno, polja koja se umanjuju *moraju ostati nenegativna*, tj. mora vrijediti $75 - t \geq 0$, $115 - t \geq 0$ i $90 - t \geq 0$, odnosno $t \leq 75$, $t \leq 115$ i $t \leq 90$. Stoga je najveća moguća promjena transporta t_{max} jednaka *najmanjoj* od vrijednosti koja se nalaze u poljima koja se umanjuju, odnosno $t_{max} = \min\{75, 115, 90\} = 75$. Preraspodjelom t_{max} količinskih jedinica transporta nove vrijednosti transporta u poljima koja se mijenjaju iznose $x_{3,1} = t_{max} = 75$, $x_{3,3} = 75 - t_{max} = 0$, $x_{2,3} = 5 + t_{max} = 80$, $x_{2,2} = 115 - t_{max} = 40$, $x_{1,2} = 10 + t_{max} = 85$ i $x_{1,1} = 90 - t_{max} = 15$. Novo dopustivo rješenje prikazano je u sljedećoj tabeli:

	P_1	P_2	P_3	P_4
S_1	15 / 8	85 / 9	4	6
S_2	6	40 / 9	80 / 5	3
S_3	75 / 5	6	7	65 / 4

Troškovi transporta za ovakvo rješenje iznose

$$Z = 8 \cdot 15 + 9 \cdot 85 + 9 \cdot 40 + 5 \cdot 80 + 5 \cdot 75 + 4 \cdot 65 = 2280$$

Vidimo da je rješenje poboljšano. Prazna polja su sada $x_{1,3}$, $x_{1,4}$, $x_{2,1}$, $x_{2,4}$, $x_{3,2}$ i $x_{3,3}$, pa treba naći koeficijente $d_{1,3}$, $d_{1,4}$, $d_{2,1}$, $d_{2,4}$, $d_{3,2}$ i $d_{3,3}$. Na analogan način kao u prethodnoj iteraciji, nalazimo

$$d_{1,3} = c_{1,3} - c_{1,2} + c_{2,2} - c_{2,3} = 4 - 9 + 9 - 5 = -1$$
$$d_{1,4} = c_{1,4} - c_{1,1} + c_{3,1} - c_{3,4} = 6 - 8 + 5 - 4 = -1$$
$$d_{2,1} = c_{2,1} - c_{2,2} + c_{1,2} - c_{1,1} = 6 - 9 + 9 - 8 = -2$$
$$d_{2,4} = c_{2,4} - c_{2,2} + c_{1,2} - c_{1,1} + c_{3,1} - c_{3,4} = 3 - 9 + 9 - 8 + 5 - 4 = -4$$
$$d_{3,2} = c_{3,2} - c_{3,1} + c_{1,1} - c_{1,2} = 6 - 5 + 8 - 9 = 0$$
$$d_{3,3} = c_{3,3} - c_{3,1} + c_{1,1} - c_{1,2} + c_{2,2} - c_{2,3} = 7 - 5 + 8 - 9 + 9 - 5 = 5$$

Objasnimo, na primjer, kako je određen koeficijent $d_{1,4}$. Da bismo uvećali transport preko polja $x_{1,4}$, moramo umanjiti transport ili preko polja $x_{1,2}$, ili preko polja $x_{1,1}$ (radi balansa u skladištu S_1). Pokušamo li da umanjimo transport preko polja $x_{1,2}$, moraćemo prvo uvećati transport preko polja $x_{2,2}$, a zatim umanjiti transport preko polja $x_{2,3}$. Međutim, nakon ovoga ne možemo učiniti ništa da sačuvamo balans kod potrošača P_3. Slijedi da umanjenje transporta preko polja $x_{1,2}$ *nije bila dobra ideja*. S druge strane, pokušamo li umanjiti transport preko polja $x_{1,1}$, tada moramo uvećati transport preko polja $x_{3,1}$, a zatim smanjiti transport preko polja $x_{3,4}$, čime se ciklus zatvara. Ovim smo pronašli traženi poligonalni ciklus.

Sada je najnegativniji relativni koeficijent troškova $d_{2.4}$, tako da ćemo izvršiti preraspodjelu transporta na polje $x_{2,4}$. Ovom polju odgovara ciklus $x_{2,4} \to x_{2,2} \to x_{1,2} \to x_{1,1} \to x_{3,1} \to x_{3,4} \to x_{2,4}$. Polja koja se pri tome umanjuju su $x_{2,2}$, $x_{1,1}$ i $x_{3,4}$, pa je maksimalna količina koja se smije preraspodijeliti jednaka $t_{max} = \min\{x_{2,2}, x_{1,1}, x_{3,4}\} = \min\{40, 15, 65\} = 15$. Nove vrijednosti transporta u poljima gdje se vrši preraspodjela transporta iznose $x_{2,4} = t_{max} = 15$, $x_{2,2} = 40 - t_{max} = 25$, $x_{1,2} = 85 + t_{max} = 100$, $x_{1,1} = 15 - t_{max} = 0$, $x_{3,1} = 75 + t_{max} = 90$ i $x_{3,4} = 65 - t_{max} = 50$. Nakon izvršene preraspodjele, dobija se novo dopustivo rješenje prikazano u sljedećoj tabeli:

	P_1	P_2	P_3	P_4
S_1	8	100 / 9	4	6
S_2	6	25 / 9	80 / 5	15 / 3
S_3	90 / 5	6	7	50 / 4

Troškovi transporta za ovakvo rješenje iznose

$$Z = 9 \cdot 100 + 9 \cdot 25 + 5 \cdot 80 + 3 \cdot 15 + 5 \cdot 90 + 4 \cdot 50 = 2220$$

Nastavljamo dalje. Prazna polja su sada $x_{1,1}$, $x_{1,3}$, $x_{1,4}$, $x_{2,1}$, $x_{3,2}$ i $x_{3,3}$. Relativni koeficijenti troškova koji odgovaraju ovim poljima su sljedeći:

$$d_{1,1} = c_{1,1} - c_{1,2} + c_{2,2} - c_{2,4} + c_{3,4} - c_{1,4} = 8 - 9 + 9 - 3 + 4 - 5 = 4$$
$$d_{1,3} = c_{1,3} - c_{1,2} + c_{2,2} - c_{2,3} = 4 - 9 + 9 - 5 = -1$$
$$d_{1,4} = c_{1,4} - c_{1,2} + c_{2,2} - c_{2,4} = 6 - 9 + 9 - 3 = 3$$
$$d_{2,1} = c_{2,1} - c_{2,4} + c_{3,4} - c_{1,4} = 6 - 3 + 4 - 5 = 2$$
$$d_{3,2} = c_{3,2} - c_{3,4} + c_{2,4} - c_{2,2} = 6 - 4 + 3 - 9 = -4$$
$$d_{3,3} = c_{3,3} - c_{3,4} + c_{2,4} - c_{3,2} = 7 - 4 + 3 - 5 = 1$$

Najnegativniji relativni koeficijent troškova je sada $d_{3,2}$, a odgovarajućem polju $x_{3,2}$ na koje ćemo preraspodijeliti transport odgovara ciklus $x_{3,2} \rightarrow x_{3,4} \rightarrow x_{2,4} \rightarrow x_{2,2} \rightarrow x_{3,2}$. Polja koja se umanjuju su $x_{3,4}$ i $x_{2,2}$, te je $t_{max} = \min\{50, 25\} = 25$, a nove vrijednosti transporta su $x_{3,2} = 25$, $x_{3,4} = 50-25 = 25$, $x_{2,4} = 15+25 = 40$ i $x_{2,2} = 25-25 = 0$. Ovim se dobija novo dopustivo rješenje kao u sljedećoj tabeli.

	P_1	P_2	P_3	P_4
S_1	8	100 / 9	4	6
S_2	6	9	80 / 5	40 / 3
S_3	90 / 5	25 / 6	7	25 / 4

Sa ovakvim rješenjem, troškovi transporta iznose

$$Z = 9 \cdot 100 + 5 \cdot 80 + 3 \cdot 40 + 5 \cdot 90 + 6 \cdot 25 + 4 \cdot 25 = 2120$$

Može se primijetiti da se troškovi transporta polako ali sigurno poboljšavaju (smanjuju). Relativni koeficijenti troškova za prazna polja sada iznose:

$$d_{1,1} = c_{1,1} - c_{1,2} + c_{3,2} - c_{3,1} = 8 - 9 + 6 - 5 = 0$$
$$d_{1,3} = c_{1,3} - c_{1,2} + c_{3,2} - c_{3,4} + c_{2,4} - c_{2,3} = 4 - 9 + 6 - 4 + 3 - 5 = -5$$
$$d_{1,4} = c_{1,4} - c_{1,2} + c_{2,2} - c_{2,4} = 6 - 9 + 6 - 4 = -1$$
$$d_{2,1} = c_{2,1} - c_{2,4} + c_{3,4} - c_{1,4} = 6 - 3 + 4 - 5 = 2$$
$$d_{2,2} = c_{2,2} - c_{2,4} + c_{3,4} - c_{3,2} = 9 - 3 + 4 - 6 = 4$$
$$d_{3,3} = c_{3,3} - c_{3,4} + c_{2,4} - c_{3,2} = 7 - 4 + 3 - 5 = 1$$

Najnegativniji relativni koeficijent troškova je $d_{1,3}$, dok odgovarajućem polju $x_{1,3}$ odgovara ciklus $x_{1,3} \rightarrow x_{1,2} \rightarrow x_{3,2} \rightarrow x_{3,4} \rightarrow x_{2,4} \rightarrow x_{2,3} \rightarrow x_{1,3}$. Polja koja se umanjuju su $x_{1,2}$, $x_{3,4}$ i $x_{2,3}$, tako da imamo $t_{max} = \min\{100, 25, 80\} = 25$. Nove vrijednosti transporta nakon obavljenih izmjena postaju $x_{1,3} = 25$, $x_{1,2} = 100-25 = 75$, $x_{3,2} = 25+25 = 50$, $x_{3,4} = 25-25 = 0$, $x_{2,4} = 40+25 = 65$ i $x_{2,3} = 80-25 = 55$, tako da se dobija rješenje kao u sljedećoj tabeli:

	P_1	P_2	P_3	P_4
S_1	8	75 / 9	25 / 4	6
S_2	6	9	55 / 5	65 / 3
S_3	90 / 5	50 / 6	7	4

Troškovi transporta koji odgovaraju ovom rješenju iznose

$$Z = 9 \cdot 75 + 4 \cdot 25 + 5 \cdot 55 + 3 \cdot 65 + 5 \cdot 90 + 6 \cdot 50 = 1995$$

Nađimo ponovo relativne koeficijente troškova za prazna polja:

$$d_{1,1} = c_{1,1} - c_{1,2} + c_{3,2} - c_{1,4} = 8 - 9 + 6 - 5 = 0$$
$$d_{1,4} = c_{1,4} - c_{1,3} + c_{2,3} - c_{2,4} = 6 - 4 + 5 - 3 = 4$$
$$d_{2,1} = c_{2,1} - c_{2,3} + c_{1,3} - c_{1,2} + c_{3,2} - c_{3,1} = 6 - 5 + 4 - 9 + 6 - 5 = -3$$
$$d_{2,2} = c_{2,2} - c_{2,3} + c_{1,3} - c_{1,2} = 9 - 5 + 4 - 9 = -1$$
$$d_{3,3} = c_{3,3} - c_{3,2} + c_{1,2} - c_{1,3} = 7 - 6 + 9 - 4 = 6$$
$$d_{3,4} = c_{3,4} - c_{3,2} + c_{1,2} - c_{1,3} + c_{2,3} - c_{2,4} = 4 - 6 + 9 - 4 + 5 - 3 = 5$$

Najnemativniji relativni koeficijent troškova sada je $d_{2,1}$. Polju $x_{2,1}$ na koje ćemo preraspodijeliti transport odgovara ciklus $x_{2,1} \rightarrow x_{2,3} \rightarrow x_{1,3} \rightarrow x_{1,2} \rightarrow x_{3,2} \rightarrow x_{3,1} \rightarrow x_{2,1}$. Ovdje se umanjuju polja $x_{2,3}$, $x_{1,2}$ i $x_{3,1}$, tako da je $t_{max} = \min\{55, 75, 90\} = 55$. Nove vrijednosti transporta su $x_{2,1} = 55$, $x_{2,3} = 55-55 = 0$, $x_{1,3} = 25+55 = 80$, $x_{1,2} = 75-55 = 20$, $x_{3,2} = 50+55 = 105$ i $x_{3,1} = 90-55 = 35$. Ovim se dobija rješenje prikazano u sljedećoj tabeli:

	P_1	P_2	P_3	P_4
S_1	8	20 / 9	80 / 4	6
S_2	55 / 6	9	5	65 / 3
S_3	35 / 5	105 / 6	7	4

Ovom rješenju odgovaraju troškovi transporta

$$Z = 9 \cdot 20 + 4 \cdot 80 + 6 \cdot 55 + 3 \cdot 65 + 5 \cdot 35 + 6 \cdot 105 = 1830$$

Ovo rješenje je zapravo *optimalno* (odnosno, *dostigli smo optimalno rješenje*), ali to još ne možemo znati *prije nego što odredimo relativne koeficijente troškova* i uvjerimo se da među njima *nema negativnih*. Zaista, relativni koeficijenti troškova za prazna polja sada iznose

$$d_{1,1} = c_{1,1} - c_{1,2} + c_{3,2} - c_{1,4} = 8 - 9 + 6 - 5 = 0$$
$$d_{1,4} = c_{1,4} - c_{1,2} + c_{3,2} - c_{3,1} + c_{2,1} - c_{2,4} = 6 - 9 + 6 - 5 + 6 - 3 = 1$$
$$d_{2,2} = c_{2,2} - c_{2,1} + c_{3,1} - c_{3,2} = 9 - 6 + 5 - 6 = 2$$
$$d_{2,3} = c_{2,3} - c_{2,1} + c_{3,1} - c_{3,2} + c_{1,2} - c_{1,3} = 5 - 6 + 5 - 6 + 9 - 4 = 3$$
$$d_{3,3} = c_{3,3} - c_{3,2} + c_{1,2} - c_{1,3} = 7 - 6 + 9 - 4 = 6$$
$$d_{3,4} = c_{3,4} - c_{3,1} + c_{2,1} - c_{2,4} = 4 - 5 + 6 - 3 = 2$$

Pošto su svi relativni koeficijenti troškova *veći ili jednaki nuli*, to pokazuje da je ovo rješenje *zaista optimalno*. Dakle, optimalno rješenje postavljenog problema glasi

$x_{1,1} = 0$	$x_{1,2} = 20$	$x_{1,3} = 80$	$x_{1,4} = 0$
$x_{2,1} = 55$	$x_{2,2} = 0$	$x_{2,3} = 0$	$x_{2,4} = 65$
$x_{3,1} = 35$	$x_{3,2} = 105$	$x_{3,3} = 0$	$x_{3,4} = 0$

S druge strane, činjenica da su neki od ovih koeficijenata nule (tačnije $d_{1,1}$) ukazuje da dobijeno rješenje *nije i jedino optimalno rješenje*. Drugo optimalno rješenje moglo bi se dobiti preraspodjelom transporta preko polja $x_{1,1}$ čime bi se dobio *drugačiji transport*, ali *iste ukupne cijene* (ovim bismo zapravo dobili ono optimalno rješenje koje smo dobili kao početno rješenje primjenom Vogelovog aproksimativnog metoda). Zapravo, tako nađena dva optimalna rješenja predstavljaju *bazna optimalna rješenja*. Kako su transportni problemi specijalni slučajevi problema linearnog programiranja, i kod njih vrijedi da ako optimalno rješenje nije jedinstveno, tada zapravo postoji *beskonačno mnogo optimalnih rješenja*. Tako, pored ovako nađena dva *bazna* optimalna rješenja, upravo razmotreni transportni problem ima još i beskonačno mnogo drugih *nebaznih* optimalnih rješenja, koja se mogu izraziti kao *linearne kombinacije* nađena dva bazna optimalna rješenja.

Može se primijetiti da je u ovom primjeru bio potreban relativno veliki broj iteracija (tačnije pet) da se dostigne optimalno rješenje. Ovo je posljedica činjenice da smo krenuli od početnog rješenja koje je *dosta daleko od optimalnog*. Da smo umjesto od rješenja dobijenog metodom sjeverozapadnog ugla krenuli od rješenja dobijenog metodom minimalnih jediničnih troškova, bila bi potrebna *samo jedna iteracija* da se dostigne optimalno rješenje, dok smo Vogelovim metodom slučajno odmah na početku dobili rješenje koje je optimalno. Naravno, to ne mora uvijek biti tako.

Mada je metod stepping-stone *konceptualno vrlo jednostavan*, on je *dosta nepraktičan za ručni rad*. Naime, vidjeli smo da je kod ovog metoda prvo potrebno na dosta mučan način za svako od polja preko kojih se ne vrši transport (tj. za koje je $x_{i,j} = 0$) odrediti relativne koeficijente troškova $d_{i,j}$, formiranjem pravougaonog poligona koji počinje u razmatranom polju a ima ostale vrhove isključivo u poljima preko kojih se vrši transport. U nastavku će biti prikazan alternativni metod, poznat kao ***MODI*** (od *modified distribution*) ***metod***, koji ove koeficijente računa na *mnogo jednostavniji način*, *pogotovo pri ručnom računanju*. Inače, nakon što se odrede svi koeficijenti $d_{i,j}$, dalji postupak unapređivanja rješenja prema *MODI* metodu teče na isti način kao i kod metoda stepping-stone (dakle, jedina je razlika u tome kako se računaju ovi koeficijenti).

MODI metod zasniva se na *teoriji dualnosti* i *dualnom problemu transportnog problema*. Naime, neka je dat opći oblik transportnog problema

$$\arg\min Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{i,j} x_{i,j}$$

p.o.

$$\sum_{j=1}^{n} x_{i,j} = a_i\,,\ \ i = 1\,..\,m$$

$$\sum_{i=1}^{m} x_{i,j} = b_j\,,\ \ j = 1\,..\,n$$

$$x_{i,j} \geq 0\,,\ \ i = 1\,..\,m\,,\ \ j = 1\,..\,n$$

Lako je vidjeti da dualni problem ovog problema glasi

$$\arg\max W = \sum_{i=1}^{m} a_i u_i + \sum_{j=1}^{n} b_j v_j$$

p.o.

$$u_i + v_j \leq c_{i,j}\,,\ \ i = 1\,..\,m\,,\ \ j = 1\,..\,n$$

Ovdje su iz praktičnih razloga dualne promjenljive označene sa u i v a ne sa y, jer se pokazuje pogodnim imati drugačije oznake za dualne promjenljive koje su spregnute sa prvom odnosno sa drugom skupinom ograničenja (u literaturi na južnoslovenskim prostorima se umjesto u i v često susreću oznake r i k, od *red* i *kolona*). Primijetimo također da *nema ograničenja na znak dualnih promjenljivih.*

Uvođenjem izravnavajućih (dopunskih) promjenljivih $d_{i,j}$, $i = 1\,..\,m$, $j = 1\,..\,n$ ovaj problem se može napisati u sljedećem obliku, u kojem su sva ograničenja tipa jednakosti:

$$\arg\max W = \sum_{i=1}^{m} a_i u_i + \sum_{j=1}^{n} b_j v_j$$

p.o.

$$u_i + v_j + d_{i,j} = c_{i,j}\,,\ \ i = 1\,..\,m\,,\ \ j = 1\,..\,n$$

$$d_{i,j} \geq 0$$

Ove izravnavajuće dualne promjenljive su namjerno označene baš sa $d_{i,j}$, jer se lako pokazuje da njihove vrijednosti *nisu ništa drugo nego relativni koeficijenti troškova* koji nam trebaju. Zaista, ovi koeficijenti imaju istovjetno značenje kao i *negirani koeficijenti u posljednjem redu simpleks tabele pri primjeni simpleks algoritma* (negacija je posljedica činjenice da je u svojoj tipičnoj formi transportni problem *problem minimizacije*). S druge strane, znamo i da u svakoj iteraciji simpleks algoritma komplementarne vrijednosti dualnih promjenljivih također dobijamo negacijom koeficijenata u posljednjem redu simpleks tabele. Odavde zaključujemo da su relativni koeficijenti troškova koji odgovaraju ma kojem baznom rješenju istovjetni sa komplementarnim dualnim promjenljivim koje odgovaraju tom istom baznom rješenju. Pri tome, kao što znamo iz opće teorije dualnosti, te vrijednosti dualnih promjenljivih *neće biti dopustive* (tj. barem jedna dualna promjenljiva $d_{i,j}$ neće zadovoljavati uvjet $d_{i,j} \geq 0$), *osim kad je rješenje optimalno*. Ovo se savršeno slaže sa onim što znamo da ako je makar jedan relativni koeficijent troškova $d_{i,j}$ negativan, *tada rješenje nije optimalno*.

Iz izloženog slijedi da ukoliko imamo neki način za efikasno računanje osnovnih dualnih promjenljivih u_i, $i = 1\,..\,m$ i v_j, $j = 1\,..\,n$ koje odgovaraju nekom tekućem baznom rješenju, tada je veoma lako izračunati relativne koeficijente troškova $d_{i,j}$ koristeći formulu

$$d_{i,j} = c_{i,j} - u_i - v_j$$

jer su oni zapravo istovjetni *tekućim vrijednostima izravnavajućih dualnih promjenljivih*. Ostaje samo problem *kako efikasno računati vrijednosti* u_i, $i = 1\,..\,m$ i v_j, $j = 1\,..\,n$. Odgovor na ovo pitanje slijedi iz *svojstva oslabljene komplementarnosti* prema kojem je $x_{i,j}\, d_{i,j} = 0$ za sve $i = 1\,..\,m$ i $j = 1\,..\,n$. Na osnovu ovoga slijedi

$$x_{i,j} \neq 0 \ \ \Rightarrow\ \ d_{i,j} = 0 \ \ \Rightarrow\ \ u_i + v_j = c_{i,j}$$

Dakle, za svako polje preko kojeg se vrši transport, zbir odgovarajućih dualnih promjenljivih u_i i v_j jednak je jediničnoj cijeni transporta $c_{i,j}$. Pri tome, s obzirom na činjenicu da je jedno ograničenje u polaznom transportnom problemu *uvijek suvišno*, to za posljedicu ima da će u dualnom problemu *jedna od osnovnih dualnih promjenljivih imati proizvoljnu vrijednost*. Kako u slučaju da bazno rješenje nije degenerirano polja preko kojih se vrši transport ima $m+n-1$ dok osnovnih dualnih promjenljivih ima $m+n$, slijedi da *imamo taman dovoljno jednačina da se mogu odrediti sve osnovne dualne promjenljive*, uz pretpostavku da *vrijednost jedne od njih odaberemo proizvoljno* (situaciju kada je bazno rješenje degenerirano razmotrićemo kasnije). Pri tome se lako može pokazati da ta proizvoljnost u konačnici *ne utiče na izračunate vrijednosti* $d_{i,j}$, odnosno izračunate vrijednosti za $d_{i,j}$ će biti iste *kakva god da se uzme ta proizvoljna vrijednost* (stoga se obično uzima da je ta proizvoljna vrijednost jednaka nuli). Pored toga, dobijeni sistem jednačina za određivanje osnovnih dualnih promjenljivih je *veoma lagan za riješiti*, zbog toga što se jednačine uvijek mogu razmatrati *takvim redoslijedom da je u svakoj sljedećoj jednačini koja se razmatra uvijek jedna promjenljiva poznata*, pa se ona druga lako odredi, bez potrebe da se razmatra čitav sistem jednačina.

Razmotrimo, na primjer, konkretan transportni problem koji smo razmatrali na početku, za koji je dobijeno sljedeće početno dopustivo rješenje prema metodu sjeverozapadnog ugla:

	P_1	P_2	P_3	P_4
S_1	90 8	10 9	4	6
S_2	6	115 9	5 5	3
S_3	5	6	75 7	65 4

Na osnovu gore izloženog, osnovne dualne promjenljive koje odgovaraju ovom rješenju zadovoljavaju sljedeći sistem jednačina:

$u_1 + v_1 = 8$

$u_1 + v_2 = 9$

$u_2 + v_2 = 9$

$u_2 + v_3 = 5$

$u_3 + v_3 = 7$

$u_3 + v_4 = 4$

Jednu od promjenljivih možemo postaviti na nulu. Najbolje je da to bude promjenljiva koja se javlja u što je god moguće više jednačina. U našem primjeru, u_1, u_2, u_3, v_2 i v_3 se javljaju u po dvije jednačine, a nijedna od promjenljivih se ne javlja u više od dvije jednačine. Jednu od njih, recimo u_1, postavićemo na nulu, tako da dalje imamo:

$v_1 = 8 - u_1 = 8 - 0 = 8$

$v_2 = 9 - u_1 = 9 - 0 = 9$

$u_2 = 9 - v_2 = 9 - 9 = 0$

$v_3 = 5 - u_2 = 5 - 0 = 5$

$u_3 = 7 - v_3 = 7 - 5 = 2$

$v_4 = 4 - u_3 = 4 - 2 = 2$

Dakle, dobili smo

$u_1 = 0 \quad u_2 = 0 \quad u_3 = 2 \quad v_1 = 8 \quad v_2 = 9 \quad v_3 = 5 \quad v_4 = 2$

Stoga, u skladu sa prethodnim izlaganjem, dalje imamo:

$d_{1,3} = c_{1,3} - u_1 - v_3 = 4 - 0 - 5 = -1$

$d_{1,4} = c_{1,4} - u_1 - v_4 = 6 - 0 - 2 = 4$

$d_{2,1} = c_{2,1} - u_2 - v_1 = 6 - 0 - 8 = -2$

$d_{2,3} = c_{2,4} - u_2 - v_4 = 3 - 0 - 2 = 1$

$d_{3,1} = c_{3,1} - u_3 - v_1 = 5 - 2 - 8 = -5$

$d_{3,2} = c_{3,2} - u_3 - v_2 = 6 - 2 - 9 = -5$

Upoređujući sa ovim koeficijentima koje smo ranije dobili stepping-stone metodom, vidimo da su dobijene *iste vrijednosti*, samo uz *mnogo manje posla*.

U praksi se vrijednosti dualnih promjenljivih nalaze još lakše, *bez eksplicitnog raspisivanja jednačina za njihovo određivanje*. Vidjeli smo da nam, nakon što se odredi početno dopustivo rješenje, vrijednosti potreba i zaliha više ne trebaju. Umjesto njih, u transportnoj tabeli uvešćemo novi red i kolonu za vrijednosti dualnih promjenljivih, kao na sljedećoj slici:

	P_1	P_2	P_3	P_4	u_i
S_1	90 / 8	10 / 9	4	6	
S_2	6	115 / 9	5 / 5	3	
S_3	5	6	75 / 7	65 / 4	
v_j					

Na nulu postavljamo onu dualnu promjenljivu koja odgovara redu i koloni *sa najviše nenultih transporta*. Takvi su ovdje S_1, S_2, S_3, P_2 i P_3, pa se nasumice odlučujemo za S_1, odnosno stavljamo $u_1 = 0$. Dalje samo *gledamo polja u tabeli preko kojih se vrši transport* i *računamo ono što se u tom trenutku može računati*. Kako mora biti $u_1 + v_1 = 8$, odmah popunjavamo $v_1 = 8$. Isto tako, mora biti $u_1 + v_2 = 9$, tako da stavljamo $v_2 = 9$. Dalje možemo iskoristiti $v_2 + u_2 = 9$ da dobijemo $u_2 = 0$. Nakon toga koristimo $u_2 + v_3 = 5$ odakle slijedi $v_3 = 5$. Sada iz $u_3 + v_3 = 7$ slijedi $u_3 = 2$, da bi konačno iz $u_3 + v_4 = 4$ imali $v_4 = 2$. Sve se ovo radi *direktno nad tabelom*, *bez raspisivanja ikakvih jednačina*. Na kraju postupka tabela izgleda ovako:

	P_1	P_2	P_3	P_4	u_i
S_1	90 / 8	10 / 9	4	6	0
S_2	6	115 / 9	5 / 5	3	0
S_3	5	6	75 / 7	65 / 4	2
v_j	8	9	5	2	

Nakon što se odrede osnovne dualne promjenljive u_i i v_j, relativni koeficijenti troškova $d_{i,j}$ određuju se na isti način kao i ranije pomoću formule $d_{i,j} = c_{i,j} - u_i - v_j$. U svakom slučaju, kada se odrede svi relativni koeficijenti troškova za sva polja sa nenultim transportom, postupak se dalje nastavlja *na isti način kao kod stepping-stone metoda*, odnosno vrši se preraspodjela transporta na polje kojem odgovara najnegativniji koeficijent $d_{i,\ j}$, ukoliko takvih uopće ima (u suprotnom je rješenje optimalno) u skladu sa postupkom koji smo ranije opisali. Dakle, i ovdje je potrebno naći zatvoreni ciklus (pravougaoni poligon) koji polazi od polja na koje treba preraspodijeliti transport. Međutim, za razliku od stepping-stone metoda, ovdje je potrebno naći *samo jedan takav ciklus*, nakon što je *već određeno na koje polje se treba preraspodijeliti transport*, dok smo kod stepping-stone metoda morali tražiti *onoliko ciklusa koliko ima polja preko kojih se ne vrši transport*.

Dalje iteracije nećemo prikazivati, jer postupak teče isto kao i kod stepping-stone metoda (jedino se razlikuje izračunavanje relativnih koeficijenata troškova). Ipak, da bolje utvrdimo postupak nalaženja relativnih koeficijenata troškova pomoću *MODI* metoda, prikazaćemo još jednom tok postupka, ali ćemo ovaj put krenuti od početnog dopustivog rješenja koje je dobijeno Vogelovim aproksimativnim metodom (koje je slučajno i optimalno), a koje je prikazano u sljedećoj tabeli:

	P_1	P_2	P_3	P_4
S_1	20 / 8	9	80 / 4	6
S_2	55 / 6	9	5	65 / 3
S_3	15 / 5	125 / 6	7	4

Raspisane, jednačine za računanje osnovnih dualnih promjenljivih izgledale bi ovako:

$u_1 + v_1 = 8$
$u_1 + v_3 = 4$
$u_2 + v_1 = 6$
$u_2 + v_4 = 3$
$u_3 + v_1 = 5$
$u_3 + v_2 = 6$

Kako se promjenljiva v_1 javlja u tri jednačine, stavićemo $v_1 = 0$. Dalje je:

$u_1 = 8 - v_1 = 8 - 0 = 8$
$v_3 = 4 - u_1 = 4 - 8 = -4$
$u_2 = 6 - v_1 = 6 - 0 = 6$
$v_4 = 3 - u_2 = 3 - 6 = -3$
$u_3 = 5 - v_1 = 5 - 0 = 5$
$v_2 = 6 - u_3 = 6 - 5 = 1$

Alternativno, sve možemo uraditi i direktno nad tabelom. Kako u koloni P_1 imamo najviše nenultih transporta, odmah stavljamo $v_1 = 0$, nakon čega iz uvjeta $u_1 + v_1 = 8$, $u_2 + v_1 = 6$ i $u_3 + v_1 = 5$ odmah čitamo i popunjavamo $u_1 = 8$, $u_2 = 6$ i $u_3 = 5$. Sada uvjet $u_1 + v_3 = 4$ daje $v_3 = -4$, uvjet $u_2 + v_4 = 3$ daje $v_4 = -3$, dok uvjet $u_3 + v_2 = 6$ daje $v_2 = 1$. Ovo je prikazano u sljedećoj tabeli:

	P_1	P_2	P_3	P_4	u_i
S_1	20 / 8	9	80 / 4	6	8
S_2	55 / 6	9	5	65 / 3	6
S_3	15 / 5	125 / 6	7	4	5
v_j	0	1	–4	–3	

U svakom slučaju, dobili smo da je

$u_1 = 8 \quad u_2 = 6 \quad u_3 = 5 \quad v_1 = 0 \quad v_2 = 1 \quad v_3 = -4 \quad v_4 = -3$

Stoga je:

$d_{1,2} = c_{1,2} - u_1 - v_2 = 9 - 8 - 1 = 0$
$d_{1,4} = c_{1,4} - u_1 - v_4 = 6 - 8 - (-3) = 1$
$d_{2,2} = c_{2,2} - u_2 - v_2 = 9 - 6 - 1 = 2$
$d_{2,3} = c_{2,3} - u_2 - v_3 = 5 - 6 - (-4) = 3$
$d_{3,3} = c_{3,3} - u_3 - v_3 = 7 - 5 - (-4) = 6$
$d_{3,4} = c_{3,4} - u_3 - v_4 = 4 - 5 - (-3) = 2$

Činjenica da su svi nađeni $d_{i,j}$ nenegativni ukazuje da se ovo rješenje ne može unaprijediti, odnosno da je optimalno. S druge strane, činjenica da su neki od ovih koeficijenata nule (tačnije $d_{1,2}$) ukazuje da dobijeno rješenje nije i jedino optimalno rješenje (što smo već i ranije znali). Drugo optimalno rješenje moglo bi se dobiti preraspodjelom transporta preko polja $x_{1,2}$ čime bi se dobio drugačiji transport, ali iste ukupne cijene.

Bitno je uočiti da svi postupci za unapređenje rješenja koriste *samo operacije sabiranja i oduzimanja*. Posebno je bitno da se nigdje *ne javlja operacija dijeljenja*. Slijedi da ako je *početno dopustivo bazno rješenje cjelobrojno*, tada će ovi postupci obavezno dovesti do *cjelobrojnog optimalnog rješenja*. Također, vidljivo je i da će svi postupci za nalaženje početnog dopustivog baznog rješenja *uvijek dati cjelobrojno rješenje kad god su svi kapaciteti skladišta i potrošača cjelobrojni*. Odavde dolazimo do vrlo važnog zaključka da *transportni problemi kod kojih su svi kapaciteti cjelobrojni uvijek imaju cjelobrojno optimalno rješenje*. Drugim riječima, pri cjelobrojnim kapacitetima, *eventualna dopunska ograničenja na cjelobrojnost promjenljivih uopće ne mijenjaju način rješavanja problema* (jer će optimalno rješenje biti tada cjelobrojno samo po sebi). Treba ipak napomenuti da u slučajevima kada optimalno rješenje *nije jedinstveno*, čak i u slučajevima kada su svi kapaciteti cjelobrojni pored baznih cjelobrojnih optimalnih rješenja (koja moraju postojati) postoji i beskonačno mnogo drugih *alternativnih nebaznih optimalna rješenja koja tipično nisu cjelobrojna* (osim u iznimnim situacijama). Zbog toga transportne probleme ne treba nikada rješavati *metodima unutrašnje tačke* za rješavanje općih problema linearnog programiranja, jer ovi metodi u takvim slučajevima tipično pronađu neko optimalno rješenje koje nije cjelobrojno.

Implementacioni aspekti rješavanja transportnih problema

Bitno je naglasiti da iako su metodi za rješavanje transportnih problema intuitivno veoma jasni i jednostavni za ručni rad, *programiranje ovih metoda na računaru nije posve lak zadatak*, pogotovo *ukoliko želimo da program bude efikasan*. Naime, u ovim metodima javlja se nekoliko koraka koji nisu lako prevodivi u program. Stoga su tipično čak i najjednostavnije implementacije rješavanja transportnog problema na računaru dugačke više stotina linija kôda, neovisno od programskog jezika.

Jedan od najproblematičnijih koraka za programiranje je *traženje pravougaonih zatvorenih ciklusa koji povezuju nenulte transporte*. Pokažimo stoga kako se traganje za ciklusima može izvesti *algoritamski*. Postoji više pristupa ovom problemu, a svi se u suštini svode na *razne grafovske pretrage nad grafom koji povezuje nenulte transporte horizontalnim i vertikalnim granama*. Najčešće se za tu svrhu koristi ***pretraga u širinu*** odnosno ***BFS*** (Breadth First Search) ***pretraga***, pri čemu se graf ne formira eksplicitno, nego se sve manipulacije obavljaju nad transportnom tabelom.

U konktrenom slučaju ta pretraga se obavlja ovako. Neka je potrebno naći zatvoreni ciklus počev od polja na pozivciji (p, q) (tj. polja u p-tom redu i q-toj koloni) koji vodi preko polja sa nenultim transportom. Na početku, pripremimo jedan spisak (koji može biti lista, vektor ili neka druga linearna struktura podataka, čiji elementi imaju dvije komponente, nazvaćemo ih "Polje" i "Veza"). Komponenta "Polje" sadrži informacije gdje se nalazi razmatrano polje (to mogu biti koordinate polja, ili još bolje, pokazivač na strukturu koja čuva informacije pohranjene u tom polju), dok komponenta "Veza" sadrži informaciju gdje se nalazi polje sa kojeg smo u postupku pretrage došli na razmatrano polje. Na početku, spisak ima samo jedan element, čija komponenta "Polje" ukazuje na polje (p, q), dok je komponenta "Veza" prazna. Nakon toga, prvi element spiska proglasimo za referentni, a spisak proširujemo elementima koji predstavljaju sva polja sa nenultim transportom koji se nalaze u istoj koloni kao i polje (p, q) (tj. u q-toj koloni), pri čemu u svim tim elementima komponentu "Veza" postavljamo da ukazuje na polje (p, q).

Postupak se dalje nastavlja tako što se krećemo kroz spisak, stalno uzimajući sljedeći element u spisku za referentni. Ako u referentnom polju komponente "Polje" i "Veza" pripadaju istoj koloni, spisak dopunjavamo elementima koji predstavljaju polja sa nenultim transportom koji se nalaze u istom redu kao i referentno polje, a ako u referentnom polju komponente "Polje" i "Veza" pripadaju istom redu, spisak dopunjavamo elementima koji predstavljaju polja sa nenultim transportom koji se nalaze u istoj koloni kao i referentno polje. Pri tome u svim novododanim elementima, komponentu "Veza" postavljamo da ukazuje na referentno polje. Opisani postupak se ponavlja sve dok ne naiđemo ponovo na početno polje (p, q). Tada polje sa kojeg smo došli na to polje upisujemo u komponentu "Veza" koja odgovara početnom polju (prvi element u spisku). Traženi ciklus sada lako formiramo prateći kuda vode komponente "Veza" počev od početnog polja nadalje. Za potrebe programiranja, ukoliko se spisak čuva u nekoj strukturi podataka *fiksne veličine* (kao što je recimo *niz*), korisno je znati da za transportni problem formata $m \times n$ broj elemenata u spisku nikada neće preći $m + n$.

Kao primjer, neka je potrebno pronaći pravougaoni zatvoreni ciklus sa tjemenima u poljima sa nenultim transportom koja su obilježena znakom "×" u sljedećoj tabeli, počev od polja označenog podebljanim znakom "×", tj. polja na poziciji (3,5):

×			×		
		×		×	
×				**×**	×
×	×	×			
				×	

Pri ručnom radu, prostim posmatranjem (eventualno uz malo isprobavanja) lako se uočava da traženi ciklus glasi $(3, 5) \to (3, 1) \to (4, 1) \to (4, 3) \to (2, 3) \to (2, 5) \to (3, 5)$. Međutim, pokažimo sada kako se ovaj ciklus može odrediti algoritamski. Krenimo od spiska koji sadrži samo početno polje $(3, 5)$ i u kojem je pripadna komponenta "Veza" prazna:

Polje	(3,5)
Veza	–

Polje (3, 5) proglašavamo za tekuće polje. Kako se u istoj koloni nalaze i obilježena polja na pozicijama (2, 5) i (5, 5), dopunjavamo spisak elementima koja odgovaraju ovim poljima, pri čemu komponente “Veza” u ovim elementima postavljamo na (3, 5):

Polje	**(3,5)**	(2,5)	(5,5)
Veza	–	(3,5)	(3,5)

Za novo tekuće polje uzimamo polje na poziciji (2, 5). Njegova veza (3, 5) se nalazi u istoj koloni, stoga u spisak dodajemo obilježena polja koja se nalaze u istom redu kao i polje (2, 5). Postoji samo jedno takvo polje (2, 3), te ga dodajemo u spisak, a pripadnu komponentu “Veza” postavljamo na (2, 5).

Polje	(3,5)	**(2,5)**	(5,5)	(2,3)
Veza	–	(3,5)	(3,5)	(2,5)

Sada će tekuće polje biti (5, 5). S obzirom da se njegovo vezno polje (3, 5) ponovo nalazi u istoj koloni kao i tekuće polje, razmatrano označena polja u redu u kojem se nalazi polje (5, 5). Takvih polja nema, pa ništa ne dodajemo u spisak, nego prosto idemo dalje. Sljedeće tekuće polje će biti (2, 3). Njegovo vezno polje (2, 5) se nalazi u istom redu kao i ono, tako da tražimo obilježena polja u istoj koloni kao i polje (2, 3). Takvo je polje (4, 3) te ga dodajemo u spisak, a njegovo vezno polje postavljamo na (2, 3):

Polje	(3,5)	(2,5)	(5,5)	**(2,3)**	(4,3)
Veza	–	(3,5)	(3,5)	(2,5)	(2,3)

Sljedeće tekuće polje je (4, 3). U istom redu se nalaze polja (4, 1) i (4, 2) te ih dodajemo u spisak, uz postavljanje veznog polja na (4, 3), kao što je uobičajeno:

Polje	(3,5)	(2,5)	(5,5)	(2,3)	**(4,3)**	(4,1)	(4,2)
Veza	–	(3,5)	(3,5)	(2,5)	(2,3)	(4,3)	(4,3)

Prelazimo na novo tekuće polje (4, 1). U istoj koloni se nalaze polja (1, 1) i (3, 1) koja se dodaju u spisak, a njihova veza se postavlja na polje (4, 1):

Polje	(3,5)	(2,5)	(5,5)	(2,3)	(4,3)	**(4,1)**	(4,2)	(1,1)	(3,1)
Veza	–	(3,5)	(3,5)	(2,5)	(2,3)	(4,3)	(4,3)	(4,1)	(4,1)

U sljedećem koraku, tekuće polje će biti (4, 2). Kako u istoj koloni nema drugih označenih polja, ništa ne dodajemo u spisak nego idemo dalje. Novo tekuće polje je (1, 1). U istom redu se nalazi označeno polje (1, 4) koje se dodaje u spisak, sa pripadnim veznim poljem (1, 1):

Polje	(3,5)	(2,5)	(5,5)	(2,3)	(4,3)	(4,1)	(4,2)	**(1,1)**	(3,1)	(1,4)
Veza	–	(3,5)	(3,5)	(2,5)	(2,3)	(4,3)	(4,3)	(4,1)	(4,1)	(1,1)

Sljedeće tekuće polje je sada polje na poziciji (3, 1). Kako se u istom redu nalazi i početno polje (3, 5), komponentu “Veza” početnog polja postavljamo na (3, 1), čime se postupak završava:

Polje	(3,5)	(2,5)	(5,5)	(2,3)	(4,3)	(4,1)	(4,2)	(1,1)	**(3,1)**	(1,4)
Veza	(3,1)	(3,5)	(3,5)	(2,5)	(2,3)	(4,3)	(4,3)	(4,1)	(4,1)	(1,1)

Traženi ciklus sada prosto očitavamo prateći vezna polja. Polazeći od početnog polja (3,5), njegovo vezno polje je (3,1). Vezno polje polja (3,1) je polje (4,1), itd. Ciklus je očitan kada prateći veze ponovo naiđemo na početno polje.

Izloženo objašnjenje algoritma za nalaženje ciklusa popunjava nedostajuću kariku koja je bila neophodna za programiranje stepping-stone metoda, s obzirom da se ostali koraci metoda lako prevode u odgovarajuće algoritamske korake. Međutim, ukoliko želimo da implementacija bude *efikasna*, potrebno je izvršiti još nekoliko trikova. Na prvom mjestu, *mnogo se vremena gubi* na pronalaženje *koja se polja sa nenultim transportom nalaze u istom redu odnosno koloni sa posmatranim poljem* ukoliko se to traganje vrši *prostom sekvencijalnom pretragom reda ili kolone*, pogotovo ako je tablica transporta velika (tim prije što se u svakom redu ili koloni tipično nalazi mnogo manje polja sa nenultim transportom u odnosu na ukupan broj elemenata u redu ili koloni). Da bi pretraga bila efikasnija, sve elemente sa nenultim transportom unutar jednog reda ili kolone treba čuvati u *nekoj strukturi podataka sa brzim vezama između elemenata*, recimo u *povezanim listama*. U suštini, neophodno je da se za svako polje sa nenultim transportom mogu *veoma brzo naći njemu susjedna polja sa nenultim transportom* u istom redu ili koloni. Efektivno, to znači da bi svako polje sa nenultim transportom interno u sebi trebalo sadržavati neku vrstu *veze* na njemu susjedna polja (te veze mogu biti *koordinate susjednih polja* ili *pokazivači na susjedna polja*). Naravno, sve te veze se moraju na neki način *ažurirati* kada god dođe do preraspodjele transporta. Uglavnom, bez takvih veza, implementacija algoritma *ne može biti efikasna*. Za slučaj transportnih problema dimenzije $n \times n$ razlika u brzini izvršavanja dobrih i loših implementacija je tipično sa faktorom oko n, pa čak i više, sve do faktora reda n^2 (za slučaj vrlo loših implementacija).

Još jedna stvar za koju nije baš očigledno kako je efikasno izvesti algoritamski je određivanje dualnih promjenljvih u_i, $i = 1\,..\,m$ i v_j, $j = 1\,..\,m$ u *MODI* metodu. Pri ručnom radu, čovjek prilično lako uočava *kojim redom* treba računati ove promjenljive da se iskoriste vrijednosti koje su prethodno računate, ali taj redoslijed *nije lako izraziti algoritamski*. Naravno, jedna od mogućnosti je da za svako polje (p, q) sa nenultim transportom za koje još uvijek nisu izračunati i u_p i v_q ispitujemo da li je jedna od te dvije vrijednosti već izračunata i da onda izračunamo onu drugu iz relacije $u_p + v_q = c_{p,q}$. Mada je ovaj postupak lako izvodljiv, on je prilično neefikasan.

Znatno efikasniji postupak može se ponovo izvesti *BFS pretragom*. Kako se ma koja od osnovnih dualnih promjenljivih može odabrati da bude jednaka nuli, bez umanjenja općenitosti možemo uzeti da je to promjenljiva v_1, tj. stavićemo $v_1 = 0$. Nakon toga se obavlja BFS pretraga preko polja sa nenultim transportom na istovjetan način kao pri traženju ciklusa, samo što pretraga započinje počev od nekog fiktivnog polja koje se nalazi u *prvoj koloni*, ali negdje *izvan tabele*. Obično se uzima da je to fiktivno polje u fiktivnom *nultom redu* prve kolone, tj. polje na (fiktivnoj) poziciji $(0, 1)$. Inače, da smo zbog nekog razloga umjesto v_1 htjeli da nula bude neka druga promjenljiva v_q, započeli bismo pretragu od fiktivnog polja izvan tabele u q-toj koloni, a da smo htjeli da nula bude neka promjenljiva u_p, započeli bismo pretragu od fiktivnog polja izvan tabele u p-tom redu. Međutim, kako to u osnovi nije bitno, odlučićemo se da pretraga započne od pozicije $(0, 1)$. Naravno, takva pretraga *neće uspjeti da pronađe ciklus*, jer *nikakvog ciklusa ovdje zapravo i nema*. Umjesto toga, pretragu prosto vršimo dok ne istestiramo *sva polja preko kojih se vrši transport*, odnosno sve dok više nije moguće dodavati nove elemente u spisak. Nakon što se formira spisak, obrađujemo redom jedan po jedan element u formiranom spisku, preskačući prvi element koji odgovara fiktivnom polju. Neka tekući element odgovara nekom polju (p, q). Njegovo vezno polje će biti ili oblika (p, r), ili oblika (r, q). Ukoliko je vezno polje oblika (p, r), to znači da u tom trenutku možemo računati promjenljivu v_q po formuli $v_q = c_{p,q} - u_p$, dok ukoliko je vezno polje oblika (r, q), to znači da trebamo računati promjenljivu u_p po formuli $u_p = c_{p,q} - v_q$. Nakon što prođemo kroz čitav spisak, sve dualne promjenljive će biti određene.

Demonstrirajmo ovaj postupak na početnom rješenju primjera koji smo ranije razmatrali koje je dobijeno Vogelovim aproksimativnim metodom. To rješenje je prikazano u sljedećoj tabeli:

	P_1	P_2	P_3	P_4
S_1	20 / 8	9	80 / 4	6
S_2	55 / 6	9	5	65 / 3
S_3	15 / 5	125 / 6	7	4

Nad ovom tabelom sad treba obaviti BFS pretragu preko polja sa nenultim transportom, polazeći od fiktivnog polja $(0, 1)$. Kako se ova pretraga obavlja na identičan način kao pretraga koju smo koristili kada smo tražili cikluse, nećemo prikazivati sve detalje postupka, nego ćemo samo prikazati spisak koji se dobija po okončanju postupka:

Polje	(0,1)	(1,1)	(2,1)	(3,1)	(1,3)	(2,4)	(3,2)
Veza	–	(0,1)	(0,1)	(0,1)	(1,1)	(2,1)	(3,1)

Sada idemo kroz spisak, redom polje po polje (ne računajući prvo fiktivno polje). Kako polje (1, 1) ima vezu (0, 1), to je $u_1 = c_{1,1} - v_1 = 8$. Dalje, kako polje (2, 1) ima također vezu (0, 1), to je $u_2 = c_{2,1} - v_1 = 6$. Istu vezu ima i sljedeće polje (3, 1), pa je $u_3 = c_{3,1} - v_1 = 5$. Sljedeće polje je (1, 3) koje za vezu ima polje (1, 1), pa je $v_3 = c_{1,3} - u_1 = -4$. Prelazimo na polje (2, 4) čija je veza polje (2, 1), tako da imamo $v_4 = c_{2,4} - u_2 = -3$. Konačno, posljednje polje u spisku je (3, 2) koje za vezu ima polje (3, 1), tako da je $v_2 = c_{3,2} - u_3 = 1$.

Posljednji problem koji se javlja pri implementaciji metoda za rješavanje transportnih problema je mogućnost da neko od baznih dopustivih rješenja bude *degenerirano*. S obzirom da pojava degeneracije pravi izvjesne probleme ne samo pri računarskoj implementaciji nego i pri *ručnom računanju*, ovom problemu kao i načinu njegovog rješavanja biće posvećen poseban odjeljak.

Problem degeneracije i njegovo rješavanje

Prilikom primjene svih metoda za unapređenje rješenja (odnosno eventualnu detekciju da je postignuto rješenje optimalno) kao što su *stepping-stone* ili *MODI* metod nastaju problemi ukoliko dopustivo rješenje posjeduje manje od $m+n-1$ nenultih transporta. Već smo rekli da za takvo dopustivo rješenje kažemo da je *degenerirano*, dok se razlika $m+n-1-p$ gdje je p broj nenultih transporta u rješenju naziva ***stepen degeneracije*** rješenja (tako da možemo govoriti o jednostruko, dvostruko, itd. degenerisanom rješenju). Degeneracija rješenja uvijek stvara izvjesne poteškoće, pri čemu se problemi povećavaju sa porastom stepena degeneracije. Degeneracija se može pojaviti odmah *na samom početku* (tj. početno dopustivo rješenje može biti degenerirano), a može se pojaviti i u *kasnijim iteracijama* (npr. neka od iteracija stepping-stone ili *MODI* metoda može po svom završetku dati degenerirano rješenje).

Razmotrimo prvo kakve probleme degeneracija stvara kod primjene *stepping-stone* metoda. Vidjeli smo da kod ovog metoda moramo za svako polje preko kojeg se ne vrši transport odrediti relativne koeficijente troškova $d_{i,j}$ formiranjem ciklusa oblika pravougaonog poligona koji počinje u razmatranom polju a ima ostale vrhove isključivo u poljima preko kojih se vrši transport. Problem je u tome što u slučaju da je razmatrano rješenje degenerisano, takav ciklus *nećemo moći napraviti* za neka od polja. Posmatrajmo na primjer sljedeći transportni problem, sa dozvoljenim rješenjem koje je također prikazano u tabeli:

	P_1	P_2	P_3	P_4	P_5	Zalihe
S_1	3	10	200 / 4	2	3	200
S_2	7	5	8	200 / 4	10	200
S_3	100 / 5	8	50 / 15	7	12	150
S_4	10	12	100 / 10	8	100 / 4	200
S_5	7	200 / 1	50 / 1	6	5	250
Potrebe	100	200	400	200	100	

U ovom primjeru imamo $m = n = 5$, pa bi dozvoljeno bazno nedegenerisano rješenje trebalo imati $m+n-1 = 9$ polja preko kojih se vrši transport. Međutim, u ovom rješenju takvih polja je 8, tako da je ovo jednostruko degenerisano rješenje. Probamo li primijeniti stepping-stone metodu, lako ćemo naći tražene cikluse krenemo li od polja $x_{1,1}$ ili $x_{1,2}$. Konkretno, to su ciklusi $x_{1,1} \rightarrow x_{1,3} \rightarrow x_{3,3} \rightarrow x_{3,1} \rightarrow x_{1,1}$ odnosno $x_{1,2} \rightarrow x_{1,3} \rightarrow x_{5,3} \rightarrow x_{5,2} \rightarrow x_{1,2}$. Međutim, probamo li takav ciklus naći polazeći od polja $x_{1,4}$, *nećemo uspjeti*. Zaista, krenemo li od ovog polja vertikalno, jedino polje na koje možemo “skočiti” (tj. preko kojeg se vrši transport) je polje $x_{2,4}$, ali sa ovog polja ne možemo nigdje dalje “skočiti” horizontalno (s obzirom da u drugom redu nema niti jednog drugog polja preko kojeg se vrši transport). S druge strane, čini se da bismo imali veći izbor ukoliko bismo sa početnog polja “skočili” horizontalno na polje $x_{1,3}$, jer odatle imamo mogućnost da nastavimo vertikalno na polje $x_{3,3}$, $x_{4,3}$ ili $x_{5,3}$. Ipak, sve je ovo uzaludno, jer *nema načina da se vratimo na početno polje* $x_{1,4}$. Zaista, na njega bismo se morali vratiti nekim “vertikalnim skokom”, a vidjeli smo da takvi skokovi iz početnog polja $x_{1,4}$ ne vode nigdje. Dakle, traženi ciklus nije moguće zatvoriti polazeći od polja $x_{1,4}$.

Lako se vidi da isti problemi nastaju i ako probamo formirati traženi ciklus polazeći od polja $x_{2,1}$, $x_{2,2}$, $x_{2,3}$, $x_{2,5}$, $x_{3,4}$, $x_{4,4}$ ili $x_{5,4}$. Ti problemi su uzrokovani činjenicom da se *u drugom redu i u četvrtoj koloni nalazi samo jedan nenulti transport*, a to je *posljedica degeneracije*. Problem se rješava tako što se na neko od polja od kojeg se ne može formirati traženi ciklus stavi *fiktivni transport infinitezimalnog iznosa* ε, sa kojim se onda postupa kao da je *nenulti transport* (iako se u svim računima takva veličina ε *tretira kao nula*). U načelu je *svejedno* na koje od takvih polja se stavlja taj fiktivni transport, jedino ga *ne smijemo* staviti na neko od polja sa kojeg je *moguće* načiniti ciklus (poput $x_{2,1}$ u navedenom primjeru), jer nam to *neće riješiti probleme uzrokvane degeneracijom*. U ovom primjeru, odlučićemo se da fiktivni transport stavimo na polje $x_{1,4}$, čime dobijamo sljedeću tabelu:

	P_1	P_2	P_3	P_4	P_5
S_1	3	10	200 / 4	ε / 2	3
S_2	7	5	8	200 / 4	10
S_3	100 / 5	8	50 / 15	7	12
S_4	10	12	100 / 10	8	100 / 4
S_5	7	200 / 1	50 / 1	6	5

Lako je vidjeti da je sada moguće zatvoriti tražene pravougaone poligone polazeći od svih polja preko kojih se ne vrši transport (nikakav, pa ni fiktivni). Na primjer, za polje $x_{2,1}$ takav poligon dat je kao $x_{2,1} \to x_{2,4} \to x_{1,4} \to x_{1,3} \to x_{3,3} \to x_{3,1} \to x_{2,1}$. Slično vrijedi i za sva ostala polja, tako da se postupak sada može nastaviti na uobičajeni način. Pri računanju, mada se za veličinu ε smatra da je različita od nule, ona se gubi pri sabiranju ili oduzimanju sa bilo čime različitim od nule (tako da je npr. $5 + \varepsilon = 5$), jedino je $0 + \varepsilon$ i dalje ε a ne 0.

Pokažimo sada kakve probleme stvara degeneracija ukoliko se koristi *MODI* metod. Razmotrimo ponovo isti primjer degeneriranog rješenja. Raspisane jednačine za određivanje osnovnih dualnih promjenljivih prema *MODI* metodu glase:

$u_1 + v_3 = 4$
$u_2 + v_4 = 4$
$u_3 + v_1 = 5$
$u_3 + v_3 = 15$
$u_4 + v_3 = 10$
$u_4 + v_5 = 4$
$u_5 + v_2 = 1$
$u_5 + v_3 = 1$

U skladu sa dosada opisanim postupkom, uzeli bismo $v_3 = 0$, jer se v_3 javlja čak u četiri jednačine. Dalje bismo imali:

$u_1 = 4 - v_3 = 4 - 0 = 4$
$u_3 = 15 - v_3 = 15 - 0 = 15$
$u_4 = 10 - v_3 = 10 - 0 = 10$
$u_5 = 1 - v_3 = 1 - 0 = 1$
$v_1 = 5 - u_3 = 5 - 15 = -10$
$v_5 = 4 - u_4 = 4 - 10 = -6$
$v_2 = 1 - u_5 = 1 - 1 = 0$

Preostale su još promjenljive u_2 i v_4, ali je preostala samo jedna jednačina $u_2 + v_4 = 4$ u kojoj se te promjenljive javljaju, tako da je iz nje same *nemoguće odrediti obje promjenljive* (ovdje ne smijemo jednu od njih dvije uzeti kao proizvoljnu, jer se *samo jedna promjenljiva iz cijelog skupa osnovnih dualnih promjenljivih smije proizvoljno odabrati*). Slijedi da nam *nedostaje još jedna jednačina* za određivanje u_2 i v_4. Do sličnog zaključka bismo došli da smo direktno bez raspisivanja jednačina izvodili postupak nad tabelom na način kako je ranije opisano. Došli bismo do sljedeće tabele, iz koje se ne vidi nikakav način da se odrede u_2 i v_4 (osim činjenice da im zbir mora biti jednak 4):

	P_1	P_2	P_3	P_4	P_5	u_i
S_1	3	10	200 / 4	2	3	4
S_2	7	5	8	200 / 4	10	(?)
S_3	100 / 5	8	50 / 15	7	12	15
S_4	10	12	100 / 10	8	100 / 4	10
S_5	7	200 / 1	50 / 1	6	5	1
v_j	–10	0	0	(?)	–6	

Ova situacija ukazuje na to da treba dodati fiktivni transport negdje u red S_2 ili u kolonu P_4, recimo na polje $x_{1,4}$. Kako je sada $x_{1,4} = \varepsilon \neq 0$, to nam daje jednačinu $u_1 + v_4 = 2$, što omogućava da možemo odrediti $v_4 = 2 - u_1 = 2 - 4 = -2$, tako da jednačina $u_2 + v_4 = 4$ konačno daje $u_2 = 4 - v_4 = 4 - (-2) = 6$. Time su određene sve dualne promjenljive i možemo nastaviti kako je uobičajeno.

U razmotrenom primjeru smo imali *jednostruko degenerirano rješenje*. U slučaju kada se javlja *višestruka degeneracije*, dodavanje jednog fiktivnog infinitezimalnog transporta *ne bi riješilo problem*. Recimo, kod stepping-stone metoda, ponovo bi se pojavila polja od kojih nije moguće zatvoriti ciklus. Stoga bi na neko od takvih polja trebalo bi ponovo dodati infinitezimalni transport, itd. Slično, kod *MODI* metoda će se morati uvoditi više infinitezimalnih transporta da bismo mogli odrediti sve osnovne dualne promjenljive. Generalno, kod oba metoda ćemo trebati uvesti *onoliko infinitezimalnih transporta koliko iznosi stepen degeneracije*. Svi ovi infinitezimalni transporti se gube u računanju sa drugim brojevima, osim pri računanju sa nulom i između sebe, tako da se uzima ne samo da je $0 + \varepsilon = \varepsilon$, nego je i $\varepsilon + \varepsilon = \varepsilon$ (ali je, ipak, $\varepsilon - \varepsilon = 0$). Treba napomenuti da iako je jednostruka degeneracija kod transportnih problema *česta u praksi*, višestruke degeneracije se ne susreću baš tako često, ali su ipak moguće. Na primjer, ukoliko svi snabdjevači imaju iste zalihe koje su ujedno jednake potrebama svakog od potrošača, ma koje bazno dopustivo rješenje takvog problema biće *degenerirano sa vrlo velikim stepenom degeneracije*.

Pozicije na koje treba uvesti fiktivne infinitezimalne transporte *nikad nisu jedinstveno određene*, i u načelu je *svejedno* koje se od više mogućih pozicija odaberu, mada kasnije trajanje postupka može ovisiti od učinjenog izbora. Neki autori preporučuju da se u slučaju nedoumice *fiktivni transporti stavljaju na polja sa što je god moguće manjom cijenom*, mada ne navode nikakve jasne razloge zašto bi trebalo postupati baš tako, tako da je upitno da li je ova preporuka svrsihodna.

Prethodni primjer je ilustrirao situaciju u kojoj nam je već bilo dato degenerirano bazno dopušteno rješenje, pa smo trebali *tragati* gdje treba uvesti fiktivni infinitezimalni transport (ili više njih u slučaju većeg stepena degeneracije). Međutim, prilikom rješavanja transportnih problema, moguće pozicije gdje treba uvesti infinitezimalne transporte *otkrivaju se u toku samog postupka rješavnja*. Tako, pri određivanju početnog dopustivog rješenja bilo kojim od opisanih metoda, degeneracija će se pojaviti kad god dođemo u situaciju da u istom koraku *eliminiramo i red i kolonu* u transportnoj tabeli. U takvom slučaju, da bismo odmah odredili gdje će trebati staviti fiktivni infinitezimalni transport, nasumice izaberemo da eliminiramo *samo red* ili *samo kolonu*, a u redu ili koloni koje nismo eliminirali ostavimo infinitezimalnu količinu ε kao zalihu odnosno potražnju i nastavimo postupak (vodeći računa da se ta količina gubi u računu sa svim brojevima osim nule). Ta infinitezimalna količina će se kasnije u postupku nalaženja dopustivog rješenja rasporediti na neko polje, čime ćemo dobiti željeni fiktivni infinitezimalni transport (ili više njih, ukoliko smo u sličnu situaciju došli više puta).

Degeneracija se može pojaviti ne samo u početnom dopustivom rješenju, nego i u toku postupka unapređivanja rješenja stepping-stone ili *MODI* metodom. Konkretno, to će se desiti ukoliko u toku preraspodjele transporta koja se vrši tokom metoda, *više različitih nenultih transporta postanu jednaki nuli* (a ne samo jedan, kako je to uobičajeno). U tom slučaju, nasumice se odlučimo da samo jedan od tih transporta stvarno svedemo na nulu, a sve ostale ostavimo na infinitezimalnoj vrijednosti ε. Time ćemo nakon obavljene preraspodjele odmah dobiti tražene fiktivne infinitezimalne transporte, bez potrebe da ih naknadno postavljamo. Primjer rješavanja problema sa degeneriranim rješenjima uz detekciju pozicije infinitezimalnih transporta "u hodu" tokom rješavanja problema demonstriraćemo nešto kasnije pri razmatranju *problema raspoređivanja*, koji spadaju u specijalne slučajeve transportnih problema čija je karakteristika da su im *sva dopustiva bazna rješenja degenerirana sa vrlo visokim stepenom degeneracije*.

Kada imamo infinitezimalne transporte u transportnoj tabeli, sasvim je moguće da se pri vršenju preraspodjele transporta u nekoj iteraciji dogodi da je iznos *maksimalne količine transporta koja se može preraspodijeliti također infinitezimalan*, odnosno da je $t_{max} = \varepsilon$. U takvim slučajevima, nakon što se obavi preraspodjela, jedino što će se suštinski promijeniti je što će se *neki infinitezimalni transport prebaciti na neko drugo polje*. Niti jedan *stvarni* transport se *neće promijeniti*, niti će se promijeniti *vrijednost funkcije cilja*. U slučaju visokih stepena degeneracije, može se desiti da se u čitavom nizu uzastopnih iteracija *ne desi ništa konkretno* osim preraspodjele infinitezimalnih transporta, dok se ne uspostavi *takva konfiguracija infinitezimalnih transporta koja će omogućiti konkretno unapređenje rješenja*. To može dovesti do *velikih zastoja* i osjetnog *produžavanja trajanja algoritma*. To je i osnovni razlog zbog kojeg su problemi u kojima se javlja veliki stepen degeneracije *prilično nezgodni za rješavanje* (u suštini, identični zastoji bi se pojavili i kada bismo primijenili *klasični simpleks metod*).

Teoretski, pri visokim stepenima degeneracije bi se možda čak moglo da dogodi i *kruženje*, odnosno da se nakon niza iteracija u kojima se ništa suštinski ne dešava vratimo *na istu konfiguraciju koju smo već imali*, tako da bi tada algoritam *upao u beskonačnu petlju*. Ipak, treba napomenuti da do danas niko nije uspio da pronađe primjer u kojem bi algoritmi za rješavanje transportnih problema upali u kruženje ukoliko se koristi *standardni izbor zasnovan na Dantzigovom pravilu*, tj. preraspodjela transporta na polje sa *najnegativnijim relativnim koeficijentom troškova* (za neka druga pravila izbora poznati su slučajevi kruženja), ali *niko nije ni dokazao da takvi primjeri ne postoje*. Stoga je najsigurnije da, ukoliko u svim situacijama kada trebamo da se odlučimo između više ravnopravnih mogućnosti (npr. koji ćemo transport svesti na nulu a koje ćemo ostaviti na infinitezimalnom nivou), izbor vršimo *nasumično* a ne po nekom *fiksnom* pravilu (npr. po pravilu da uvijek na nulu svodimo *prvi od više mogućih transporta koji se mogu svesti na nulu*). Naime, to nam daje garanciju da čak i ako bi se desilo da upadnemo u "začarani krug", zahvaljujući slučajnom izboru *prije ili kasnije ćemo se iz tog kruga* "*izvući*" i algoritam će "pobjeći" iz beskonačne petlje. Uglavnom, sa vrlo velikom sigurnošću možemo reći da problem kruženja kod transportnih problema *praktično ne postoji*, mada *veliki zastoji* u izvršavanju uzrokovani visokom degeneracijom i *nisu tako rijetki*.