

Metode za rješavanje statičkog problema raspoređivanja u okruženju nesrodnih strojeva

Lucija Ulaga

Voditelj: Dr. sc. Marko Đurasević

Mentor: Prof. dr. sc. Domagoj Jakobović

SADRŽAJ

- Problem
- Cilj
- Iscrpna pretraga
 - Branch and bound
- Pravilo raspoređivanja
 - ATC
- Heuristike
 - GRASP
 - GRASP+PR
 - GRASP+PR+ILS
- VNS
- VND
- Metaheuristike
 - Optimizacija kolonijom mrava
 - Simulirano kaljenje
 - Tabu pretraživanje
- Rezultati
- Zaključak

PROBLEM

- Rasporediti poslove unutar nekog sustava po resursima s ciljem minimiziranja vremena izvođenja ili drugih troškova
- n poslova i m strojeva
- oznaka j za pojedini posao, i za stroj
- **NP** težak problem

PROBLEM

- Parametri poslova:
 - w_j - težina posla j
 - p_{ij} - vrijeme izvođenja posla j na stroju i
 - r_j - vrijeme dolaska posla j u sustav
 - d_j - rok izvršavanja posla j
- Okruženje strojeva:
 - **Nesrodni** strojevi
- Dodatna ograničenja:
 - Ograničenje vremena dolaska u sustav
- Uvjeti raspoređivanja:
 - **Statičko** raspoređivanje

PROBLEM

- Kriterij optimizacije:
 - **Minimizacija** ukupnog težinskog kašnjenja

$$TWT(s) = \sum_{j=1}^n w_j * \max(C_j - d_j, 0)$$

C_j - vrijeme završetka posla j
 s - rješenje

CILJ

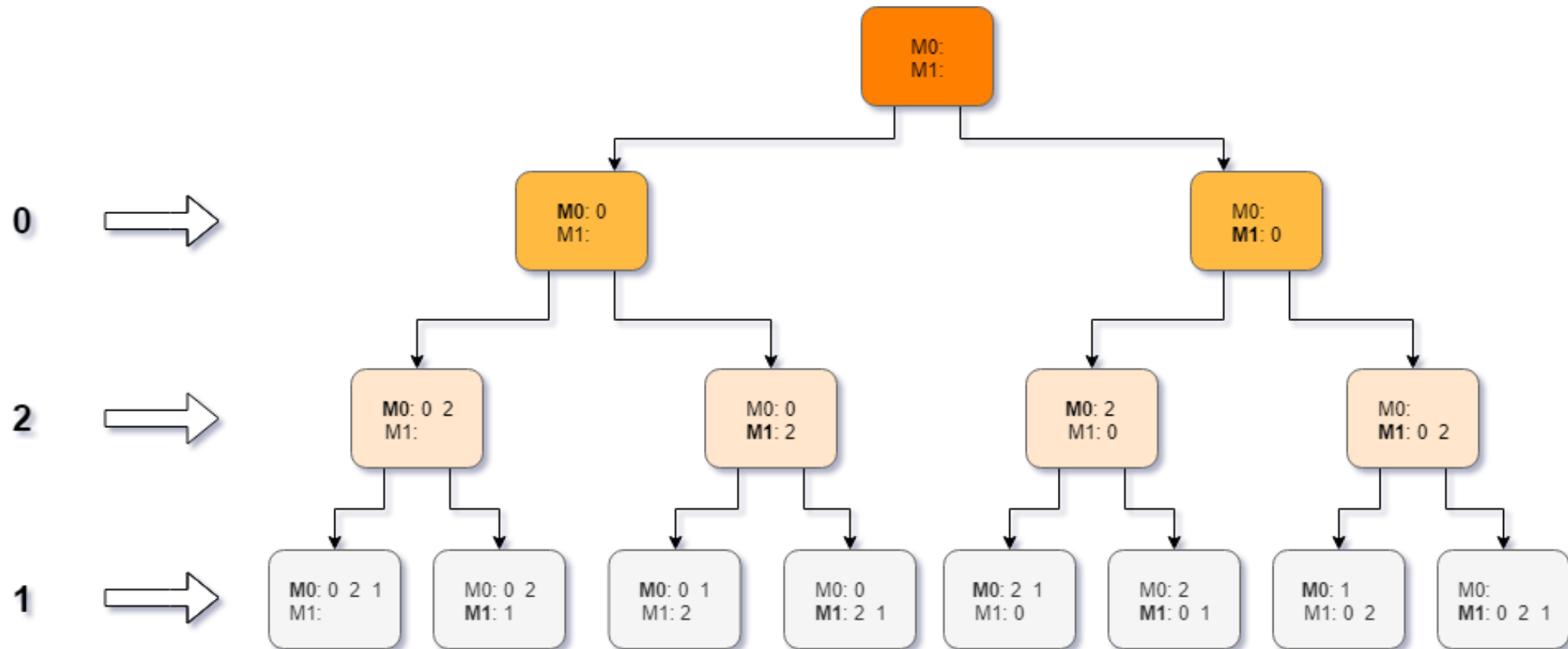
- Pregled različitih metoda i postupaka za rješavanje problema statičkog raspoređivanja (iscrpno pretraživanje, heuristike, metaheuristike)
- Usporedba s postojećim metodama iz literature (genetski algoritam, pravilo raspoređivanja)

ISCRPNA PRETRAGA

- Pronalazak optimalnih rješenja
- Branch and bound
 - Rekurzivan obilazak svih mogućih kombinacija raspoređivanja poslova po strojevima
 - **Gornja granica** - ATC ili pomoću genetskog algoritma
 - Evaluacija parcijalnog rasporeda - ako je dobrota iznad gornje granice, odbacuje cijelo podstablo

BRANCH AND BOUND

Primjer: permutacija poslova [0 2 1] na dva stroja, M0 i M1



PRAVILO RASPOREĐIVANJA

- *ATC - Apparent Tardiness Cost*
- Pravilo raspoređivanja za rješavanje problema minimizacije ukupnog težinskog kašnjenja
- U svakom trenutku kada stroj postane slobodan:
 - izračun **prioriteta** svih neraspoređenih poslova
 - odabir posla s najvećim prioritetom

ATC

- Računa prioritete u trenutcima kada postoji slobodan resurs

$$I_{i,j}(t) = \frac{w_j}{p_{i,j}} \exp \left[-\frac{\max(d_j - p_{i,j} - t, 0)}{k\bar{p}} \right]$$

t trenutno vrijeme

$I_{i,j}(t)$ prioritet raspoređivanja posla j na stroj i u trenutku t

k slobodni koeficijent

\bar{p} prosječno trajanje izvođenja poslova koji još nisu raspoređeni

HEURISTIKE

- Jednostavne heuristike koje koriste operatore lokalne pretrage
- Lokalni operatori:
 - Izmjena **redoslijeda** izvođenja poslova
 - Izmjena **dodjele** poslova resursima
- Metode:
 - GRASP, GRASP+PR, GRASP+PR+ILS
 - VNS
 - VND

GRASP

- *Greedy Randomized Adaptive Search Procedure*
- **Dvije faze:**
 - Konstrukcija rješenja
 - Poboljšavanje rješenja

GRASP

KONSTRUKCIJA RJEŠENJA

- Iterativno - smjesti posao na jedan od strojeva
- **Pohlepan** odabir stroja za svaki neraspoređeni posao (najmanja vrijednost funkcije dobrote)
- **Nasumičan** odabir jednog od neraspoređenih poslova

GRASP

LOKALNA PRETRAGA

- Uklanjanje posla i **umetanje** na neko drugo mjesto u rasporedu
- Pretraživanje susjedstva, odabir najboljeg susjeda
- Pronađen bolji susjed od trenutnog rješenja - postavlja se za trenutno rješenje i pretraga se nastavlja
- Pretraga staje kada se dođe do lokalnog optimuma

GRASP+PR

- *Path Relinking*
- **Memorija:**
 - spremanje najboljih (elitnih) rješenja
 - veličina određena parametrom *esize*
- Istražuje puteve u lokalnom prostoru između dva rješenja:
 - s_0 - **početno** rješenje (lokalna pretraga)
 - s_g - **ciljno** rješenje (nasumično odabrano elitno rješenje)

GRASP+PR+ILS

- *Iterated Local Search*
- **Perturbacije** najboljeg rješenja
 - Dvije faze: destrukcija i konstrukcija
- **Destrukcija** - uklanjanje d nasumično odabranih poslova iz rasporeda
- **Konstrukcija** - pohlepno vraćanje uklonjenih poslova u raspored

VNS

- *Variable Neighbourhood Search*
- Koristi **tri** susjedstva različitih složenosti:
 - Umetanje posla s jednog stroja na drugi
 - Zamjena poslova na jednom nasumičnom stroju
 - Zamjena poslova na različitim nasumičnim strojevima
- Početno rješenje:
 - Generirano nasumično

VNS

- U svakoj iteraciji primjenjuje se **jedno** susjedstvo
- Teži se korištenju najbrže metode generiranja susjedstva:
 - Kada je pronađeno bolje rješenje - koristi prvo, najjednostavnije susjedstvo
 - Kada nije pronađen bolji susjed - koristi iduće susjedstvo po složenosti (ciklički)

VND

- *Variable Neighbourhood Search*
- Korištenje više lokalnih pretraga, deterministički redoslijed (koriste se sve, jedna za drugom)
- **Modifikacija** rješenja
- Početno rješenje pohlepno:
 - Posao na onaj stroj na kojem ima najmanje vrijeme izvršavanja

VND

- Dvije lokalne pretrage:
 - Umetanje poslova
 - Zamjena poslova
- Zaustavljanje postupka kada je pronađen lokalni optimum s obzirom na **obje** lokalne pretrage
- Modifikacija lokalnog optimuma

VND

MODIFIKACIJE

- IG (*Iterated greedy*) - jednako postupku perturbacije
- NSP (*No Same Place*) - umetanja posla na neki drugi stroj
- VIR (*Virtual*) - umetanja posla na bilo koji stroj, uključujući onaj na kojem se posao nalazi

METAHEURISTIKE

- Općenite, jednostavne metaheuristike
- Metode:
 - Optimizacija kolonijom mrava (ACO)
 - Simulirano kaljenje (SA)
 - Tabu pretraživanje (TS)

OPTIMIZACIJA KOLONIJOM MRAVA

- Ponašanje mrava u prirodi:
 - pronalazak najkraćeg puta do hrane
 - feromoni, isparavanje
- Mrav:
 - prolazi put od praznog do potpunog rasporeda
 - u svakom čvoru donosi odluku o raspoređivanju temeljenu na vjerojatnosti
- Vjerojatnost:
 - **feromoni** - dobrota puta
 - **vidljivost** - pohlepna heuristika, vrijeme izvođenja

OPTIMIZACIJA KOLONIJOM MRAVA

- Ažuriranje feromona:
 - isparavanje nakon svakog mrava
 - put kojim je išao najbolji mrav
- Mrav gradi raspored kroz dvije faze:
 1. Alokacija stroja za svaki posao
 2. Permutacija poslova, redoslijed kojim se poslovi raspoređuju na strojeve
- Lokalna pretraga:
 - Nasumično odaberi 5% poslova i odaberi drugačiji stroj
 - Zamjena dva nasumično odabrana posla na istom stroju

SIMULIRANO KALJENJE

- Motivirano kaljenjem metala, radi s jednim rješenjem
- Parametar T :
 - Predstavlja **temperature**
- U svakoj iteraciji se pretražuje susjedstvo
- **Bolje** rješenje automatski se prihvaća
- **Lošije** rješenje se prihvaća s određenom vjerojatnosti

SIMULIRANO KALJENJE

VJEROJATNOST

- Smanjuje kako se temperatura smanjuje

$$P(s') = \min\left(1, e^{\frac{f(s) - f(s')}{T}}\right)$$

f funkcija dobrote
 s trenutno rješenje
 s_0 novo rješenje
 T temperatura

SIMULIRANO KALJENJE

- Plan hlađenja:

$$T_k = T_0 * \alpha^k$$

k - trenutni korak algoritma, iteracija

- Diverzifikacija
- Intenzifikacija
- Izbjegavanje lokalnih optimuma

TABU PRETRAŽIVANJE

- Radi s jednim rješenjem, u svakoj iteraciji bira najboljeg susjeda iz osam različitih susjedstva
- Početno rješenje - ATC
- **Tabu lista** sprječava kruženje tako da pamti zadnjih / posjećenih rješenja ili njihovih atributa
- Ako je pronađen susjed bolji od trenutnog rješenja, prihvaća se neovisno o tabu listi
- Inače, prihvaćanje najboljeg susjeda koji nije tabu, čak i ako je lošiji od trenutnog rješenja

TABU PRETRAŽIVANJE

TABU LISTA

- Pamti poteze:
 - Posao koji mijenja poziciju
 - Stroj na koji je posao premješten
 - Pozicija na koju je posao stavljen

EKSPERIMENTI

- 60 instanci problema:
 - Broj poslova – 12, 25, 50, 100
 - Broj strojeva – 3, 6, 10
- Vremensko ograničenje:
 - Iscrpna pretraga – **24 sata** za jednu instancu
 - Ostale metode – **5 minuta** za sve instance

EKSPERIMENTI

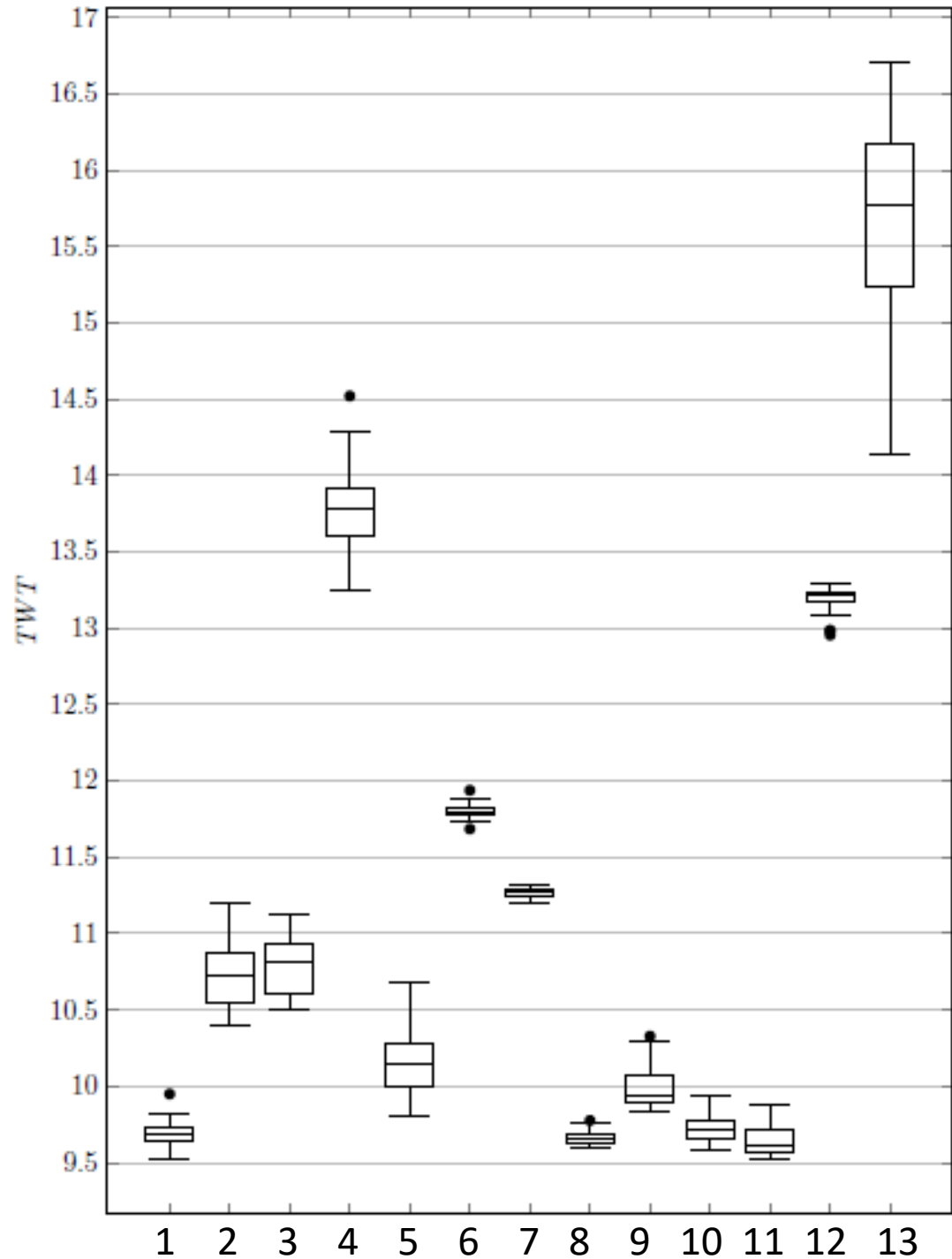
- Kriterij ocjene pojedine metode:
 - Suma dobrota po svim instancama
 - **30 pokretanja** za svaku metodu – minimum, medijan, maksimum, najbolje iz svih pokretanja
- Usporedba s genetskim algoritmom kojim su do sada dobivena najbolja rješenja
- Isprobane varijante s inicijalizacijom početnog rješenja uz ATC
- Provedena opširna optimizacija parametara

REZULTATI

ISCRPNA PRETRAGA

- Nisu pronađena rješenja bolja od GA
- 51.6% instanci – pronađeno optimalno rješenje
- 15 % instanci – prekinuta pretraga, vraćeno najbolje nađeno rješenje
- 33.3% instance – nije pronađeno nikakvo rješenje
- Nedostatak iscrpne pretrage – ne nalazi rješenja za velike instance

metoda	min	med	max	best
SA+ATC	9.5274	9.6165	9.8831	9.4475
GRASP+PR+ILS+ATC	9.5905	9.6601	9.7759	9.4691
GA	9.5278	9.6772	9.9493	9.4314
SA	9.5759	9.7097	9.9366	9.4737
GRASP+PR+ILS	9.8316	9.9342	10.3283	9.5146
VNS	9.8035	10.1437	10.6849	9.4639
VND+IG	10.4000	10.7483	11.2046	9.8218
VND+VIR	10.5048	10.8245	11.1249	10.0155
GRASP+PR+ATC	11.1970	11.2700	11.3247	11.1589
GRASP+ATC	11.6852	11.8005	11.9383	11.5929
TS	12.9549	13.2244	13.2892	12.5611
ATC	-	13.3821	-	-
VND+NSP	13.2499	13.7840	14.5203	12.2617
ACO	14.1345	15.7891	16.7021	11.6569
GRASP+PR	25.1047	27.1770	28.8285	20.6035
GRASP	110.1150	116.5735	121.2370	89.9420



- 1 - GA
- 2 - VND+IG
- 3 - VND+VIR
- 4 - VND+NSP
- 5 - VNS
- 6 - GRASP+ATC
- 7 - GRASP+PR+ATC
- 8 - GRASP+PR+ILS+ATC
- 9 - GRASP+PR+ILS
- 10 - SA
- 11 - SA+ATC
- 12 - TS
- 13 - ACO

ZAKLJUČAK

- Provedeni eksperimenti daju opširan **pregled metoda** za rješavanje statičkog raspoređivanja u okruženju nesrodnih strojeva
- **Simulirano kaljenje** uz ATC daje izvrsne rezultate
- **GRASP+PR+ILS** uz ATC vrlo blizu rezultatima GA
- GRASP loši rezultati – neprilagođenost problem
- VND i VNS prihvatljivi rezultati
- Optimizacija kolonijom mrava i tabu pretraživanje ostavljaju prostora za unaprjeđenja u daljnjem radu
- Korištenje **ATC**-a za generiranje **početnog rješenja**

KRAJ

- Hvala na pažnji!

LITERATURA

Jean-Paul Arnaout, Ghaith Rabadi, i Rami Musa. A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times. *Journal of Intelligent Manufacturing*, 21:693–701, 12 2009.

J. Behnamian, M. Zandieh, i S.M.T. Fatemi Ghomi. Parallel-machine scheduling problems with sequence-dependent setup times using an aco, sa and vns hybrid algorithm. *Expert Systems with Applications*, 36(6):9637 – 9644, 2009.

Marco Dorigo, Mauro Birattari, i Thomas Stützle. Ant colony optimization. *Computational Intelligence Magazine, IEEE*, 1:28–39, 12 2006.

J. Du i J.Y. Leung. Minimizing total tardiness on one machine is np-hard. *Mathematics of Operations Research*, 15:483–494, 1990.

Luis Fanjul-Peyro i Rubén Ruiz. Iterated greedy local search methods for unrelated parallel machine scheduling. *European Journal of Operational Research*, 207(1):55 – 69, 2010.

Thomas A. Feo i Mauricio G. C. Resende. Greedy randomized adaptive search procedures. *J. Global Optimization*, 6:109–133, 1995.

Paola Festa i Mauricio G. C. Resende. An annotated bibliography of grasp—part ii: Applications. *International Transactions in Operational Research*, 16(2):131–172, 2009.

- Johnson D. Garey, M. Computers and intractability: A guide to the theory of NP-completeness. New York: W.H. Freeman, 1997.
- F. Glover i G. A. Kochenberger. Handbook of Metaheuristics. Kluwer Academic Publishers, New York, 2003.
- Fred Glover. Tabu search—part ii. ORSA Journal on Computing, 2(1):4–32, 1990.
- Fred Glover. Tabu search and adaptive memory programming – Advances, applications and challenges. Kluwer, 1996.
- Pierre Hansen i Nenad Mladenovic. Variable neighborhood search: Principles and applications. European Journal of Operational Research, 130:449–467, 02 2001.
- Scott Kirkpatrick, Charles Daniel Gelatt, i Mario P. Vecchi. Optimization by simulated annealing. Science, 220 4598:671–80, 1983.
- Jae-Ho Lee, Jae Min Yu, i Dong-Ho Lee. A tabu search algorithm for unrelated parallel machine scheduling with sequence- and machine-dependent setups: Minimizing total tardiness. The International Journal of Advanced Manufacturing Technology, 69, 12 2013.
- Y.H. Lee. A heuristic to minimize the total weighted tardiness with sequencedependent setups. IIE Transactions (Institute of Industrial Engineers), 29:45–52, 01 1997.
- O. C. Martin Lourenço, H. R. i T. Stützle. Iterated local search. Handbook of Metaheuristics. International Series in Operations Research and Management Science 57. Springer New York, 2003.

N. Mladenović i P. Hansen. Variable neighborhood search. *Computers and Operations Research*, 24(11):1097 – 1100, 1997.

P.S. Ow i T.E. Morton. The single machine early/tardy problems. *Management Science*, 35:177–191, 1989.

Mateus Paula, Martin Ravetti, Geraldo Mateus, i Panos Pardalos. Solving parallel machines scheduling problems with sequence-dependent setup times using variable neighbourhood search. *IMA Journal of Management Mathematics*, 18:101–115, 03 2007.

João Paulo de C. M. Nogueira, José Elias C. Arroyo, Harlem Mauricio M. Villadiego, i Luciana B. Gonçalves. Hybrid grasp heuristics to solve an unrelated parallel machine scheduling problem with earliness and tardiness penalties. *Electronic Notes in Theoretical Computer Science*, 302:53–72, 02 2014.

M. L. Pinedo. *Planning and Scheduling in Manufacturing and Services*. Springer, New York, USA, 2005.

M. L. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Springer, New York, USA, third edition, 2008.

Ribeiro C.C. Resendel M.G. GRASP with Path-Relinking: Recent Advances and Applications. In: Ibaraki T., Nonobe K., Yagiura M. (eds) *Metaheuristics: Progress as Real Problem Solvers*. *Operations Research/Computer Science Interfaces Series*, vol. 32. Springer, Boston, MA, 2005.

Rubén Ruiz i Thomas Stützle. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177:2033–2049, 2007.

D. R. Sule. *Production Planning and Industrial Scheduling: Examples, Case Studies and Applications*. CRC Press, Boca Raton, USA, second edition, 2008.

E.-G. Talbi. *Metaheuristics: From design to implementation*. Hoboken, N.J: John Wiley and Sons, 2009.

A. Vepsalainen i T.E. Morton. Priority rules for jobshops with weighted tardiness costs. *Management Science*, 33:1035–1047, 1987.

Marko Čupić. *Prirodom inspirirani optimizacijski algoritmi*. Metaheuristike. 2012.

DODATCI

PROBLEM

- Posao se mora izvršiti samo jednom na nekom od dostupnih strojeva
- Stroj u nekom trenutku može izvršavati samo jedan posao
- Kada se posao započne izvršavati na stroju, mora se na istom do kraja izvršiti
- Svi parametri poslova su deterministički i dostupni prije početka rada sustava

BRANCH AND BOUND

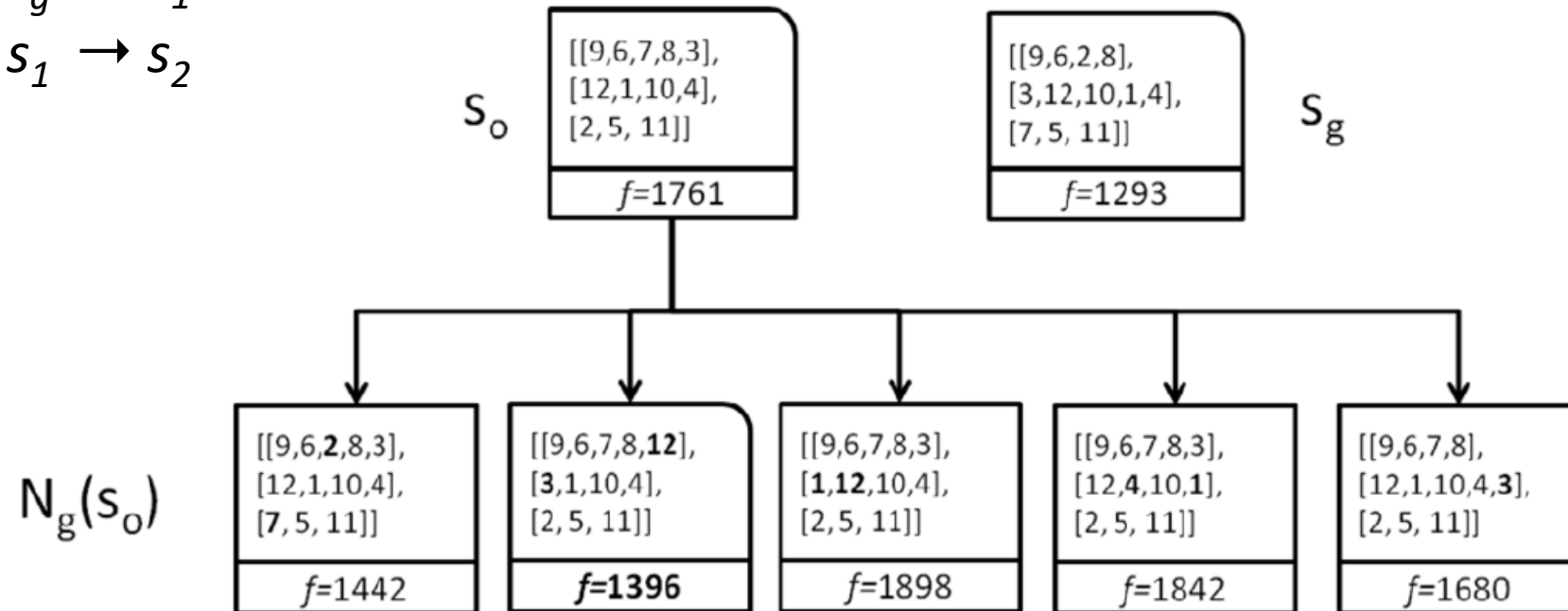
OPTIMIZACIJE

- Početna permutacija poslova:
 - sortirana po vremenu dolaska u sustav
 - najbolje rješenje nađeno genetskim algoritmom
- Ne smanjuje se prostor pretrage, već povećava vjerojatnost da je prostor boljih rješenja istražen ranije
- Ažuriranje gornje granice

GRASP+PR

- Zapravo *Mixed Path Relinking*:

- s_0 prema $s_g \rightarrow s_1$
- s_g prema $s_1 \rightarrow s_2$



Slika 1. – preuzeto iz Paulo de C. M. Nogueira et al. (2014)

SIMULIRANO KALJENJE

- Početno rješenje:
 - nasumično raspoređivanje poslova po strojevima
- Lokalna pretraga preuzeta iz VNS
- Iterativno:
 - Odabir **nasumičnog** susjeda na jedan od tri opisana načina u VNS metodi
 - Izgradnja susjedstva **zamjenom** bilo koja dva posla u rasporedu i odabir **najboljeg** susjeda

TABU PRETRAŽIVANJE

SUSJEDSTVA

1. Umetanje grupe poslova
2. Premještanje posla
3. Lančano umetanje grupa poslova
4. Lančano umetanje grupe poslova i posla
5. Lančano umetanje poslova
6. Zamjena grupa poslova
7. Zamjena poslova
8. Zamjena grupa poslova na istom stroju

OPTIMIZACIJA KOLONIJOM MRAVA

- Vjerojatnosti:

$$\Pi_{i,j}^I = \frac{(\tau_{i,j}^I)^\alpha * (\eta_{j,i})^\beta}{\sum_{l=1}^m (\tau_{l,j}^I)^\alpha * (\eta_{j,l})^\beta}$$

$$\Pi_{j,i}^{II} = \frac{\tau_{j,i}^{II}}{\sum_{l=1}^{\psi} \tau_{l,i}^{II}}$$

j - posao

i - stroj

τ - feromoni

η - vidljivost

α - utjecaj feromona

β - utjecaj vidljivosti

ψ - neraspoređeni poslovi

ρ - brzina isparavanja

- Vidljivost:

$$\eta_{j,i} = \frac{1}{p_{j,i}}$$

- Isparavanje:

$$\tau_{i,j}^I = (1 - \rho) * \tau_{i,j}^I$$

$$\tau_{j,i}^{II} = (1 - \rho) * \tau_{j,i}^{II}$$

OPTIMIZACIJA KOLONIJOM MRAVA

- Ažuriranje feromona za najboljeg mrava:

$$\tau_{i,j}^I = \tau_{i,j}^I + \Phi * \Delta\tau_{i,j}^{I,Best} \qquad \tau_{j,i}^{II} = \tau_{j,i}^{II} + \Phi * \Delta\tau_{j,i}^{II,Best}$$

$$\Delta\tau_{i,j}^{I,Best} \begin{cases} \frac{1}{TWT^{Best}}, & \text{ako je najbolji mrav rasporedio posao } j \text{ na stroj } i \\ 0, & \text{inače} \end{cases}$$

$$\Delta\tau_{j,i}^{II,Best} \begin{cases} \frac{1}{TWT^{Best}}, & \text{ako je najbolji mrav stavio posao } j \text{ na poziciju } i \\ 0, & \text{inače} \end{cases}$$

j - posao
 i - stroj
 τ - feromoni
 Φ - intenzifikacija
dobrih puteva
 TWT^{Best} - vrijednost
funkcije dobrote
najboljeg mrava

REZULTATI

OPTIMIZACIJA PARAMETARA ALGORITAMA

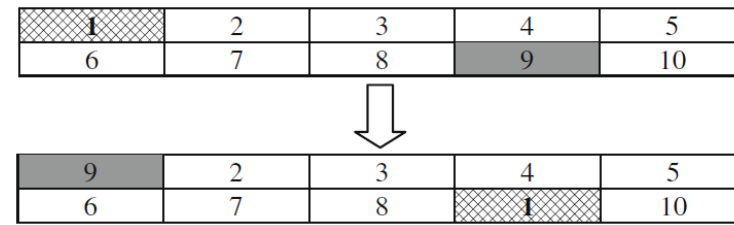
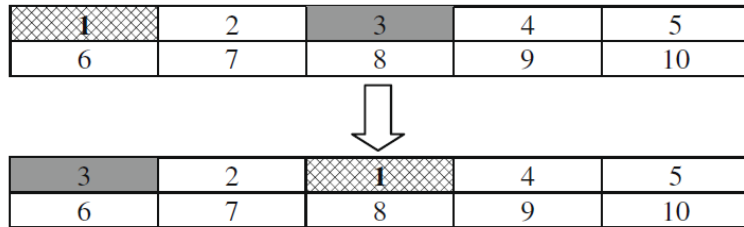
- Definiranje nekoliko vrijednosti jednog parametra i fiksiranje ostalih parametara
- **10 pokretanja** za svaku vrijednost parametra
- Odabir vrijednosti parametra – **najmanji medijan**
- Provedena opširna optimizacija parametara

REZULTATI

KORIŠTENI PARAMETRI PRI POKRETANJU

ATC	$k = 0.05$			
GRASP	$\alpha = 1$			
GRASP+ATC	$\alpha = 1$			
GRASP+PR	$\alpha = 0.9$	$esize = 0.0\dot{6}$		
GRASP+PR+ATC	$\alpha = 1$	$esize = 0.\dot{6}$		
GRASP+PR+ILS	$\alpha = 1$	$esize = 0.\dot{6}$	$d = 0.125$	$iterILS = 75$
GRASP+PR+ILS+ATC	$\alpha = 1$	$esize = 0.\dot{6}$	$d = 0.125$	$iterILS = 75$
VND+IG	$d = 0.05$			
VND+NSP	$d = 0.0\dot{6}$			
VND+VIR	$d = 0.1$			
SA	$\alpha = 0.5$	$T = 750$		
SA+ACT	$\alpha = 0.5$	$T = 200$		
TS	$l = 90$	$p = 0.1$		
ACO	$\rho = 0.01$	$\Phi = 0.7$	$iterLS = 20$	$\tau^I = 0.1$
	$\alpha = 0.8$	$k = 1$	$ants = 10$	$\tau^{II} = 0.1$

VNS



TS

