Variable Neighborhood Search for Flowshop Problem with Sequence Dependent Setup Times

Güneş Yılmaz, Ceyda Oğuz
Koç University, Istanbul, Turkey

New Challenges in Scheduling Theory
Aussois, France
April 03, 2014

Outline

- Introduction
- Literature Review
- Variable Neighborhood Search
- Computational Experiments
- Conclusion and Future Work

Introduction

- Given set N={1, …, n} of n independent jobs, processed on a set of M={1, …, m} of m machines, p_{ij}
- Consider setup times, s_{ijk} , separately from the processing time
- Permutation sequence while minimizing the makespan
- Assumptions:
  – All jobs and machines are available at time zero.
  – Processing and setup times are deterministic and known in advance.
  – A machine can process only one job; and a job is processed only on one machine at a time.
  – Preemption is not allowed.

Introduction

- The problem is denoted as $F|s_{ijk},prmu|C_{max}$ (Pinedo (2002))
- Flowshop problem with SDST while minimizing makespan is *NP*-hard (Gupta (1986))
- We propose a Variable Neighborhood Search (VNS) algorithm for the $F|s_{ijk},prmu|C_{max}$
  - Examine the performance of the various neighborhood structures, compare our results with the alternative methods and state-of-the-art

Literature Review

- Exact solutions for flowshop with sequence dependent setup times (SDST)
  - Mixed integer linear programming (MILP) models (Srikar and Ghosh (1986), Stafford and Tseng (1990), (2001), Rios-Mercado and Bard (2003))
  - Optimally solve the problem instances with about 10 jobs and few machine, when the objective is makespan

Literature Review

- Some of the existing heuristic algorithms
  - GRASP algorithm and extended the NEH heuristic (Rios-Mercado and Bard (1998))
  - Genetic and memetic algorithms (Ruiz et al. (2005)), adapted 12 algorithms
    - Genetic algorithm, simulated annealing, iterated local search, tabu search
    - NEH, GRASP, TOTAL and SETUP heuristics, TSP-based heuristic, saving index algorithm
  - Ant colony algorithm (Gajpal et al. (2006))
- State-of-the-art: iterated greedy algorithm improved with local search procedure (Ruiz and Stützle (2008))

Variable Neighborhood Search

- Searches the solution in multiple neighborhood structures and uses local search systematically

Initialization Select the set of neighborhood structures N_k, for k = 1, ..., k_{max}, that will be used in the search; find an initial solution x; choose a stopping condition;

Repeat the following sequence until the stopping condition is met:

(1)Set $k \leftarrow 1$;

(2)Repeat the following steps until $k = k_{max}$:

(a) *Shaking* Generate a point x' at random from the k^{th} neighborhood of x ($x' \in N_k(x)$);

(b) *Local search* Apply some local search method with x' as initial solution; denote with x'' the so obtained local optimum;

(c) *Move or not* If this local optimum is better than the incumbent, move there ($x \leftarrow x''$), and continue the search with N_1 ($k \leftarrow 1$); otherwise, set $k \leftarrow k+1$;

Figure 1. Steps of the basic VNS (Mladenovic and Hansen (1997))

Variable Neighborhood Search

Algorithm Initialization

- Representation of the solution as permutation of the jobs $[j_{1,} j_{2,}...,j_n]$
- NEH heuristic extended to flowshop with SDST by Rios-Mercado and Bard (1998)
  - LPT rule, suggested by Nawaz et al. (1983), is used as job selection method to construct a sequence

Variable Neighborhood Search

Neighborhood Structures and Local Search Procedure

- Well-known neighborhood structures:
  - Swap, node insertion, 2-opt
- New neighborhood structures based on setups:
  - Maximum setup time one-job insertion, maximum setup time two-job insertion, minimum setup time two-jobs insertion
- Local search based on node insertion neighborhood with steepest descent strategy

Variable Neighborhood Search

```
Function LocalSearch_NodeInsertion(x)
        improve=true;
        while (improve=true)  do
                improve=false;
                for i=1 to n do
                begins
                        remove job h from sequence  x randomly without repetition
                        x'= best sequence  obtained by inserting  job h in all possible position of
                        x;
                        if F(x')<F(x)  then
                                x=x';
                                improve=true;
                        endif
                endfor
        endwhile
        return x
end
```

Figure 3. Local search based on node insertion (Ruiz and Stützle (2007), (2008))

Variable Neighborhood Search

Acceptance Criterion

- a simple acceptance criterion
  - accept the new sequence if its makespan value is lower than the incumbent value
- a simulated annealing-like acceptance criterion
  - if *F*(candidate solution) >*F*(incumbent), but

  Random ≤ exp {(*F*(incumbent)-*F*(candidate solution)) / Temperature} then current solution ← candidate solution
  - Osman and Potts (1989)

$$Temperature = \lambda \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij}}{nxmx10}$$

Computational Experiments

Implementation of VNS

- Benchmark set generated by Taillard (1993)
  - 20, 50, 100 jobs x 5, 10, 20 machines, 200 jobs x 10, 20 machines, 500 jobs x 20 machines
  - Setup time values: 10%, 50%, 100%, 125% of processing times, denoted as SDST10, SDST50, SDST100, SDST125 (Ruiz et al. (2005))
- Code in C++, Intel Core i5-2520M 2.50GHz CPU machine
- Stopping condition based on CPU times as (*n*x*m*/2)x90 milliseconds as Ruiz and Stützle used (2008)
- Percentage deviation = ((*Solution-BestKnown*)/*BestKnown*) x 100

Computational Experiments

Implementation of VNS

- Two neighborhood structures: swap and setup dependent neighborhood structures
- Three neighborhood structures: node insertion plus two neighborhood structures
- Local search procedure based on different neighborhood structures

Computational Experiments

Table 1. Average percentage deviation from the best known solution for different neighborhood structures

Neighborhood Structure	SDST10	SDST50	SDST100	SDST125	Average
Swap - MaxSetup(1)	0.56	1.01	1.46	1.68	1.18
MaxSetup(1) - Swap	0.63	0.97	1.43	1.80	1.21
Swap - MaxSetup(2)	0.61	1.10	1.56	1.65	1.23
MaxSetup(2) - Swap	**0.52**	1.02	1.36	1.66	**1.14**
Swap - MinSetup(2)	0.56	1.02	1.48	1.80	1.21
MinSetup(2) - Swap	0.59	1.13	1.56	1.73	1.25
Swap - Insertion - MaxSetup(1)	0.60	1.02	1.61	1.77	1.25
Swap - MaxSetup(1) - Insertion	0.66	1.18	1.61	1.78	1.31
Insertion - Swap - MaxSetup(1)	0.65	1.10	1.58	1.72	1.26
Insertion - MaxSetup(1) - Swap	0.55	**0.89**	1.61	**1.60**	1.16
MaxSetup(1) - Swap - Insertion	0.58	1.10	**1.34**	1.77	1.20
MaxSetup(1) - Insertion - Swap	0.59	1.09	1.48	1.75	1.23

Computational Experiments

Table 2. Average percentage deviation from the best known solution for different neighborhood structures and local search procedures

Neighborhood Structure	Local Search	SDST10	SDST50	SDST100	SDST125	Average
MaxSetup(2) - Swap	Insertion	**0.52**	**1.02**	**1.36**	**1.66**	**1.14**
AdjacentSwap - Swap	AdjacentSwap - Insertion	0.76	1.37	1.94	2.10	1.54
Insertion(1) - Swap	Insertion	0.68	1.13	1.96	2.05	1.46
Insertion(2) - Swap	Insertion(2) - Insertion	0.99	1.86	2.61	2.82	2.07
2-opt - Swap	2-opt - Insertion	0.84	1.64	2.04	2.45	1.74
MaxSetup(2) - Swap	VND (AdjacentSwap - Insertion)	0.65	1.12	1.64	2.00	1.35

Computational Experiments

Results

- Compare with alternative methods and state-of-the-art (Ruiz and Stützle(2008)):
  - Genetic algorithm (GA) and memetic algorithm (MA) (Ruiz et al. (2005))
  - Ant colony optimization algorithm (PACO) (Rajendran and Ziegler (2004))
  - Memetic algorithm improved with the local search phase (MA_LS), iterated greedy algorithm (IG) and iterated greedy with local search phase (IG_LS), which is state-of-the-art

Computational Experiments

SDST10	**GA**	**MA**	**MA_LS**	**PACO**	**IG_RS**	**IG_RS_LS**	**VNS**
20x5	0.41	0.70	**0.08**	0.18	0.14	**0.04**	**0.08**
20x10	0.56	0.36	**0.13**	0.22	0.24	**0.04**	**0.17**
20x20	0.39	0.56	0.10	0.12	0.19	**0.04**	**0.08**
50x5	0.92	0.77	**0.30**	**0.42**	0.84	**0.27**	**0.58**
50x10	2.01	1.26	**0.81**	1.06	1.43	**0.53**	**1.03**
50x20	2.10	1.28	**0.82**	**1.01**	1.54	**0.60**	**1.18**
100x5	1.03	0.63	**0.31**	0.76	1.34	**0.33**	**0.51**
100x10	1.33	**0.90**	**0.48**	**0.77**	1.32	**0.38**	**0.95**
100x20	1.83	**1.06**	**0.82**	**1.12**	1.47	**0.54**	**1.32**
200x10	1.32	**0.65**	**0.48**	0.85	1.33	**0.32**	**0.65**
200x20	1.71	**0.87**	**0.76**	0.95	1.12	**0.38**	**0.93**
500x20	1.27	0.48	**0.43**	0.61	0.82	**0.21**	**0.43**
Average	1.24	0.79	**0.46**	0.67	0.98	**0.31**	**0.66**

Table 3. Average percentage deviation of alternative methods and the proposed VNS algorithm for SDST10

Computational Experiments

SDST125	**GA**	**MA**	**MA_LS**	**PACO**	**IG_RS**	**IG_RS_LS**	**VNS**
20x5	1.90	1.40	**0.32**	0.65	1.24	**0.30**	**0.40**
20x10	1.52	1.24	**0.37**	**0.56**	1.44	**0.36**	**0.64**
20x20	0.95	1.21	**0.24**	**0.39**	0.81	**0.19**	**0.41**
50x5	5.63	**3.48**	**1.97**	**3.67**	4.00	**2.01**	**3.80**
50x10	4.59	3.35	**1.50**	2.96	3.47	**1.54**	**2.62**
50x20	3.25	**1.63**	**1.26**	**2.06**	2.59	**1.18**	**2.06**
100x5	6.82	**3.65**	**2.52**	7.75	4.14	**1.91**	**4.08**
100x10	4.80	**2.84**	**1.94**	5.61	3.26	**1.34**	**2.89**
100x20	3.50	**2.16**	**1.50**	4.15	2.60	**1.00**	**2.23**
200x10	5.37	2.63	**2.14**	6.20	2.94	**1.17**	**2.61**
200x20	3.69	1.69	1.49	4.16	2.24	**0.76**	**1.35**
500x20	2.83	1.36	1.23	3.02	1.64	**0.52**	**0.93**
Average	3.74	2.22	**1.37**	3.43	2.53	**1.02**	**2.00**

Table 4. Average percentage deviation of alternative methods and the proposed VNS algorithm for SDST125

Conclusion and Future Work

- Proposed VNS for $F|s_{ijk},prmu|C_{max}$
  - Initialization with NEH heuristic, improve the solution with systematic changes between two neighborhood structures and employ a local search procedure based on node insertion neighborhood with steepest descent strategy
- Node insertion neighborhood structure is a powerful move operation for the local search procedure. The solution quality is mostly improved by the local search phase compared to neighborhoods.
- VNS works efficiently, compared with GA, MA, PACO, IG
  - Hybridization of the MA and the IG with local search based on node insertion improves the solution significantly (Ruiz and Stützle (2008))
- Different local search procedures, speed ups

Thank You!

References

B. N. Srikar and S. Ghosh, "A MILP Model for the n-Job, m-Stage Flowshop with Sequence Dependent Set-up Times," *International Journal of Production Research*, vol. 24, pp. 1459-1474, Nov-Dec 1986.

E. F. Stafford and F. T. Tseng, "On the Srikar-Ghosh MILP Model for the nxm SDST Flowshop Problem," *International Journal of Production Research*, vol. 28, pp. 1817-1830, Oct 1990.

F. T. Tseng and E. F. Stafford, "Two MILP Models for the n x m SDST Flowshop Sequencing Problem," *International Journal of Production Research*, vol. 39, pp. 1777-1809, May 2001.

R. Z. Rios-Mercado and J. F. Bard, "The Flow Shop Scheduling Polyhedron with Setup Times," *Journal of Combinatorial Optimization*, vol. 7, pp. 291-318, Sep 2003.

J. N. D. Gupta, "Flowshop Schedules with Sequence Dependent Setup Times," *Journal of the Operations Research Society of Japan*, vol. 29, pp. 206-219, Sep 1986.

M. Nawaz, E. E. Enscore, and I. Ham, "A Heuristic Algorithm for the m-Machine, n-Job Flowshop Sequencing Problem," *Omega-International Journal of Management Science*, vol. 11, pp. 91-95, 1983.

R. Z. Rios-Mercado and J. F. Bard, "Heuristics for the Flow Line Problem with Setup Costs," *European Journal of Operational Research*, vol. 110, pp. 76-98, Oct 1 1998.

R. Ruiz, C. Maroto, and J. Alcaraz, "Solving the Flowshop Scheduling Problem with Sequence Dependent Setup Times Using Advanced Metaheuristics - Discrete Optimization," *European Journal of Operational Research*, vol. 165, pp. 34-54, Aug 16 2005.

References

C. Rajendran and H. Ziegler, "Ant-Colony Algorithms for Permutation Flowshop Scheduling to Minimize Makespan/Total Flowtime of Jobs," *European Journal of Operational Research*, vol. 155, pp. 426-438, Jun 1 2004.

Y. Gajpal, C. Rajendran, and H. Ziegler, "An Ant Colony Algorithm for Scheduling in Flowshops with Sequence-Dependent Setup Times of Jobs," *International Journal of Advanced Manufacturing Technology*, vol. 30, pp. 416-424, Sep 2006.

R. Ruiz and T. Stutzle, "An Iterated Greedy Heuristic for the Sequence Dependent Setup Times Flowshop Problem with Makespan and Weighted Tardiness Objectives," *European Journal of Operational Research*, vol. 187, pp. 1143-1159, Jun 16 2008.

N. Mladenovic and P. Hansen, "Variable Neighborhood Search," *Computers & Operations Research*, vol. 24, pp. 1097-1100, Nov 1997.

I. H. Osman and C. N. Potts, "Simulated Annealing for Permutation Flowshop Scheduling," *Omega-International Journal of Management Science*, vol. 17, pp. 551-557, 1989.

E. Taillard, "Benchmarks for Basic Scheduling Problems," *European Journal of Operational Research*, vol. 64, pp. 278-285, Jan 22 1993.

C. Rajendran and H. Ziegler, "An Efficient Heuristic for Scheduling in a Flowshop to Minimize Total Weighted Flowtime of Jobs," *European Journal of Operational Research*, vol. 103, pp. 129-138, Nov 16 1997.

R. Ruiz and T. Stutzle, "A Simple and Effective Iterated Greedy Algorithm for the Permutation Flowshop Scheduling Problem," *European Journal of Operational Research*, vol. 177, pp. 2033-2049, Mar 16 2007.