COMPUTING NEAR-OPTIMAL SCHEDULES
JOB SHOP SCHEDULING BY LOCAL SEARCH

J K Lenstra

Rapporteur: Martin Beet

Computing near-optimal schedules

Jan Karel Lenstra
Dept of Mathematics and Computing Science
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven
The Netherlands

For many NP-hard optimization problems there are polynomial-time algorithms for finding solutions that are provably quite close to the optimum. For others certain performance guarantees are unlikely to be attained, in the sense that if there is such a good algorithm, then P = NP. We survey a number of positive and negative results on computing near-optimal solutions for machine scheduling problems, with an emphasis on multiprocessor scheduling and shop scheduling.

Job shop scheduling by local search

Jan Karel Lenstra
Dept of Mathematics and Computing Science
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven
The Netherlands

The job shop scheduling problem is one of the most difficult problem types in combinatorial optimization. Even relatively small instances of the problem are hard to solve to optimality. Recently some progress has been made in finding good approximations of the optimum by a variety of local search techniques, such as iterative improvement, simulated annealing, tabu search, variable-depth search, and genetic algorithms. We survey this work.

SHORT SCHEDULES

JAN KAREL LENSTRA

EINDHOVEN UNIVERSITY OF TECHNOLOGY

CWI, AMSTERDAM

LESLIE HALL

HAN HOOGEVEEN

COR HURKENS

ALEXANDER RINNOOY KAN

SERGEY SEVAST'JANOV

DAVID SHMOYS

ÉVA TARDOS

BART VELTMAN

DAVID WILLIAMSON

BIN PACKING

n ITEMS OF SIZE $a_1, a_2, \ldots, a_n > 0$

BINS OF CAPACITY $b > 0$

PACK ALL ITEMS IN MINIMUM # BINS

$\exists$ PACKING IN 2 BINS ?

IS $\mathcal{NP}$-COMPLETE [KARP 1972]

$\nexists$ POLYNOMIAL ALGORITHM A

WITH $\forall$ INSTANCE I : $\dfrac{A(I)}{OPT(I)} < \dfrac{3}{2}$

UNLESS $\mathcal{P} = \mathcal{NP}$

SUPPOSE $\exists$ A

- $OPT(I) \leq 2 \Rightarrow A(I) < \frac{3}{2} OPT(I) \leq 3 \Rightarrow A(I) \leq 2$
- $OPT(I) \geq 3 \Rightarrow A(I) \geq 3$

$\Rightarrow$ A ANSWERS 2-BIN QUESTION IN POLYNOMIAL TIME

$\Rightarrow \mathcal{P} = \mathcal{NP}$

$$FFD(I) \leq \frac{11}{9} OPT(I) + 4$$ [JOHNSON 1976]

$$+ 3$$ [BAKER 1985]

$$+ 1$$ [YUE 1990]

COMBINATORIAL OPTIMIZATION PROBLEM

- CLASS OF INSTANCES

- POLYNOMIAL ALGORITHM:
  INSTANCE I & OBJECT F
  $\longrightarrow$ IS F A FEASIBLE SOLUTION FOR I ?

- POLYNOMIAL ALGORITHM:
  INSTANCE I & FEASIBLE SOLUTION F
  $\longrightarrow$ NONNEGATIVE & INTEGRAL VALUE OF F

- INSTANCE I
  $\xrightarrow{?}$ FEASIBLE SOLUTION OF MINIMUM VALUE OPT(I)

ALGORITHM A :

INSTANCE $I \longrightarrow$ FEASIBLE SOLUTION OF VALUE $A(I)$

PERFORMANCE RATIO OF A:

$$R(A) = \inf\left\{ r \mid r \geq 1, \forall I : \frac{A(I)}{OPT(I)} \leq r \right\}$$

A IS POLYNOMIAL r-APPROXIMATION ALGORITHM IF

- $R(A) \leq r$
- A RUNS IN TIME POLYNOMIAL IN $|I|$

$\{A_r\}$ IS POLYNOMIAL APPROXIMATION SCHEME IF $\forall r > 1$

- $R(A_r) \leq r$
- A_r RUNS IN TIME POLYNOMIAL IN $|I|$

$\{A_r\}$ IS FULLY POLYNOMIAL APPROXIMATION SCHEME IF $\forall r > 1$

- $R(A_r) \leq r$
- A_r RUNS IN TIME POLYNOMIAL IN $|I|$ & $\frac{1}{r-1}$

IMPOSSIBILITY THEOREM

IF $\exists\, c \in \mathbb{N}$

| $\exists$ FEASIBLE SOLUTION OF VALUE $\leq c$?
| IS $\mathcal{NP}$-COMPLETE

THEN

| $\nexists$ POLYNOMIAL ALGORITHM A WITH $R(A) < \frac{c+1}{c}$
| UNLESS $\mathcal{P} = \mathcal{NP}$

SUPPOSE $\exists$ A

- $OPT(I) \leq c \implies A(I) < \frac{c+1}{c}\, OPT(I) \leq c+1 \implies A(I) \leq c$
- $OPT(I) \geq c+1 \implies$ $A(I) \geq c+$

$\implies$ A ANSWERS $\leq c$-QUESTION IN POLYNOMIAL TIME

$\implies \mathcal{P} = \mathcal{NP}$

$! \leq c \Rightarrow \not< \frac{c+1}{c}$

BIN PACKING

$! \leq 2$ [KARP 1972] $\Rightarrow \not< \frac{3}{2}$

SYMMETRIC TSP

GRAPH $G = (V, E)$, $d: E \to \mathbb{N} \cup \{0\}$

FIND HAMILTON CYCLE OF MINIMUM TOTAL WEIGHT

$! \leq 0$ [KARP 1972] $\Rightarrow \not< r$ FOR ANY $r > 1$

MULTIPROCESSOR SCHEDULING

m IDENTICAL MACHINES $M_1, \dots, M_m$

n INDEPENDENT JOBS $J_1, \dots, J_n$

PROCESSING J_j REQUIRES TIME $p_j \in \mathbb{N}$

SCHEDULE = ASSIGNMENT OF EACH JOB TO A MACHINE

LENGTH OF SCHEDULE = $\max_i \sum_{J_j \to M_i} p_j$

FIND SCHEDULE OF MINIMUM LENGTH

$R(LS) = 2 - \frac{1}{m}$	[GRAHAM 1966]
$R(LPT) = \frac{4}{3} - \frac{1}{3m}$	[GRAHAM 1969]
PAS	[HOCHBAUM & SHMOYS 1987]
NO FPAS UNLESS $\mathcal{P} = \mathcal{NP}$	[GAREY & JOHNSON 1978]

- DEPENDENT JOBS

  PRECEDENCE RELATION ON JOB SET

- NONIDENTICAL MACHINES

  PROCESSING J_j ON M_i REQUIRES TIME p_{ij}

PRECEDENCE-CONSTRAINED SCHEDULING

$R(LS) = 2 - \frac{1}{m}$ [GRAHAM 1966]

$! \leq 3 \Rightarrow \not< \frac{4}{3}$ [L& RK 1978]

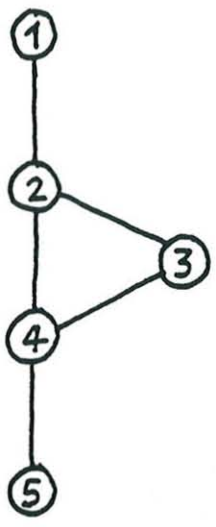
CLIQUE

GRAPH $G=(V,E)$, $k \in \mathbb{N}$

DOES G HAVE A COMPLETE SUBGRAPH ON k VERTICES?

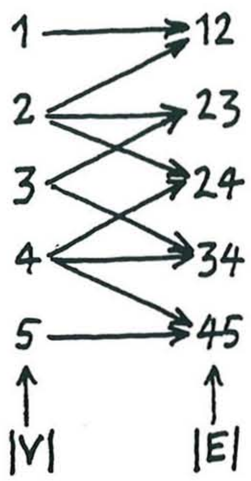
CLIQUE $\propto$ ≤ 3

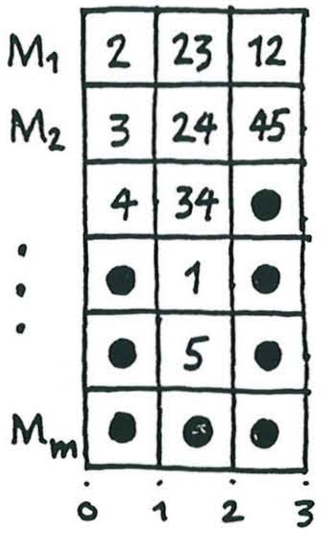
G:



m LARGE ENOUGH

$3m$ UNIT-TIME JOBS



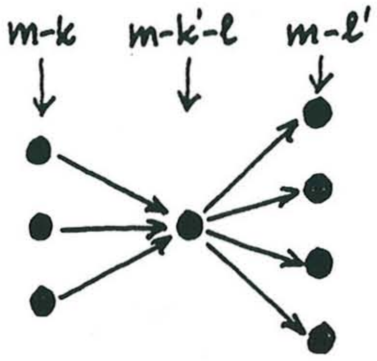


$k=3$

$\ell = \binom{k}{2}$

$k' = |V| - k$

$\ell' = |E| - \ell$



$\exists$ CLIQUE OF SIZE k $\iff$ $\exists$ SCHEDULE OF LENGTH ≤ 3

PRECEDENCE-CONSTRAINED SCHEDULING WITH COMMUNICATION DELAYS

m IDENTICAL MACHINES

n UNIT-TIME JOBS

IF $J_j \rightarrow J_k$ AND J_j & J_k ON <u>DIFFERENT</u> MACHINES

THEN UNIT-TIME DELAY BETWEEN J_j & J_k

m RESTRICTED	m UNRESTRICTED
$* \leq 3$	$* \leq 5$
$! \leq 4 \Rightarrow \not< \frac{5}{4}$	$! \leq 6 \Rightarrow \not< \frac{7}{6}$
[HOOGEVEEN, L, VELTMAN, 1992]	
$R(\text{GREEDY}) = 3$	$R(\text{LP}) = \frac{4}{3}$
[RAYWARD-SMITH]	[MUNIER, KÖNIG]
	$R_{\text{JOB DUPLICATION}}(\text{LS}) = 2$
	[PAPADIMITRIOU, YANNAKAKIS]

FOOTNOTE

m RESTRICTED	GENERAL	TREE	$m=2$
DELAYS 0	$! \leq 3$	$*$	$*$
DELAYS 1	$! \leq 4$	!	?

SCHEDULING UNRELATED MACHINES

$R(LP+M) = 2$ [L, SHMOYS, & TARDOS 1990]

$! \leqslant 3 \Rightarrow \not< \frac{4}{3}$

$! \leqslant 2 \Rightarrow \not< \frac{3}{2}$

$? \leqslant 1$ ... IS TRIVIAL

UNRELATED MACHINES

POTTS' LP-BASED ALGORITHM

(1) SOLVE LP

INPUT : p_{ij} = PROCESSING TIME OF JOB j ON MACHINE i

OUTPUT: x_{ij} = FRACTION OF JOB j ASSIGNED TO MACHINE i

$z \leq$ OPT (= MINIMUM SCHEDULE LENGTH)

$$\boxed{\begin{array}{lll} \text{MINIMIZE} & z & \\ \text{SUBJECT TO} & \sum_i x_{ij} = 1, & j = 1,\dots,n \\ & \sum_j p_{ij} x_{ij} \leq z, & i = 1,\dots,m \\ & x_{ij} \geq 0, & i = 1,\dots,m,\ j = 1,\dots,n \end{array}}$$

$\longrightarrow$ AT MOST $m-1$ SPLIT JOBS

(2) ASSIGN UNSPLIT JOBS BY ROUNDING DOWN LP-OUTPUT:

PUT FRACTION $\lfloor x_{ij} \rfloor$ OF JOB j ON MACHINE i

$\longrightarrow$ SCHEDULE OF LENGTH $\leq z \leq$ OPT

(3) ASSIGN SPLIT JOBS OPTIMALLY BY COMPLETE ENUMERATION:

CHECK $O(m^{m-1})$ POSSIBILITIES

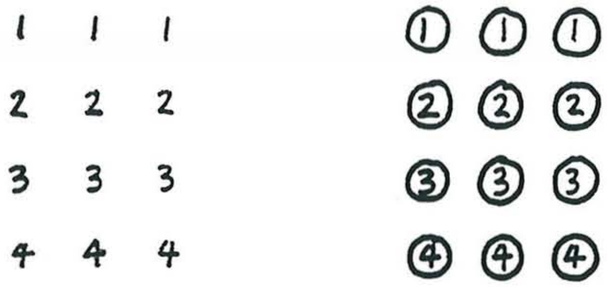
$\longrightarrow$ SCHEDULE OF LENGTH $\leq$ OPT

$R(LP) = 2$

RUNNING TIME POLYNOMIAL FOR FIXED m

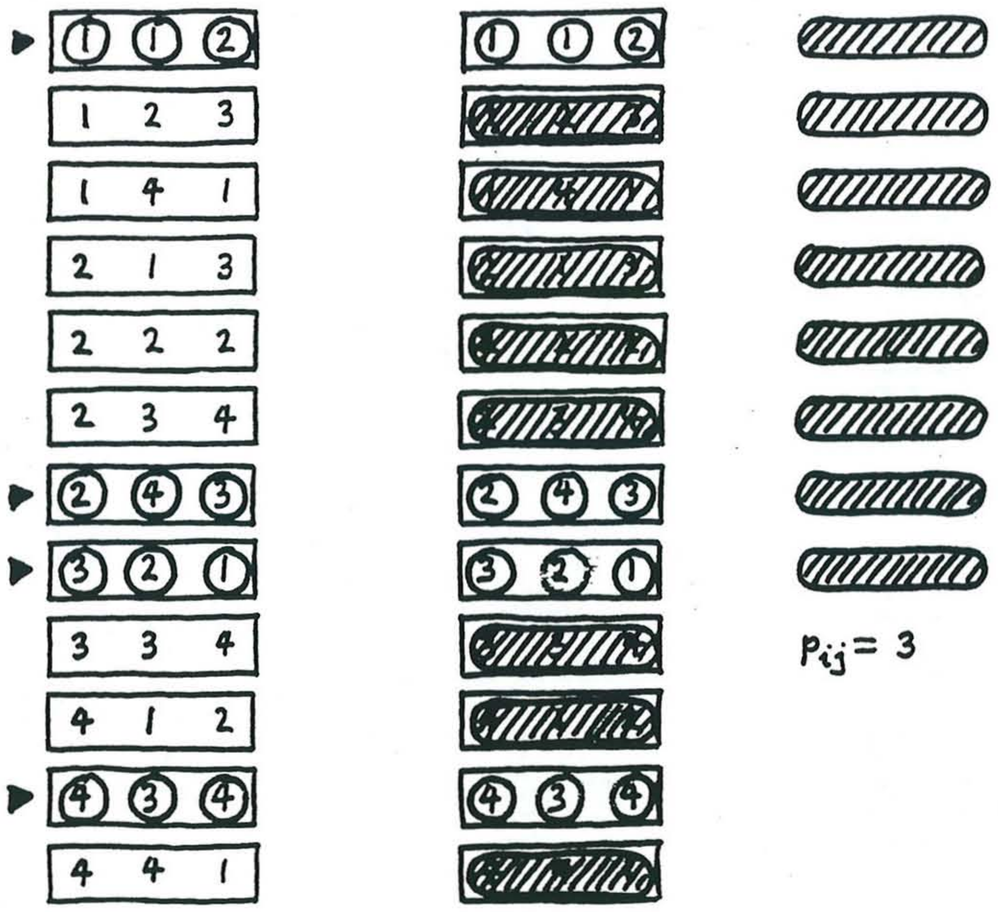
3DM $\propto$ ≤ 3

3n ELEMENTS $\rightarrow$ 3n JOBS



$$p_{ij} = \begin{cases} 1 & \text{IF } j \in \text{TRIPLE } i \\ 3 & \text{OTHERWISE} \end{cases}$$

m TRIPLES $\rightarrow$ m MACHINES & m-n DUMMY JOBS



$p_{ij} = 3$

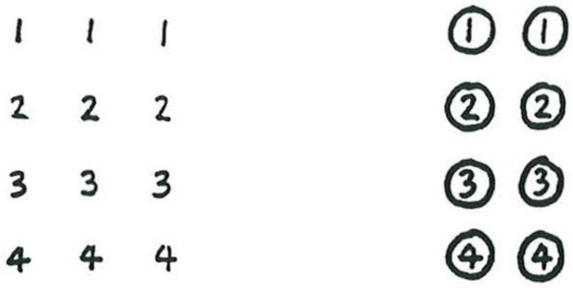
$\exists$ n TRIPLES $\Leftrightarrow$ $\exists$ SCHEDULE

CONTAINING OF LENGTH ≤ 3

3n ELEMENTS

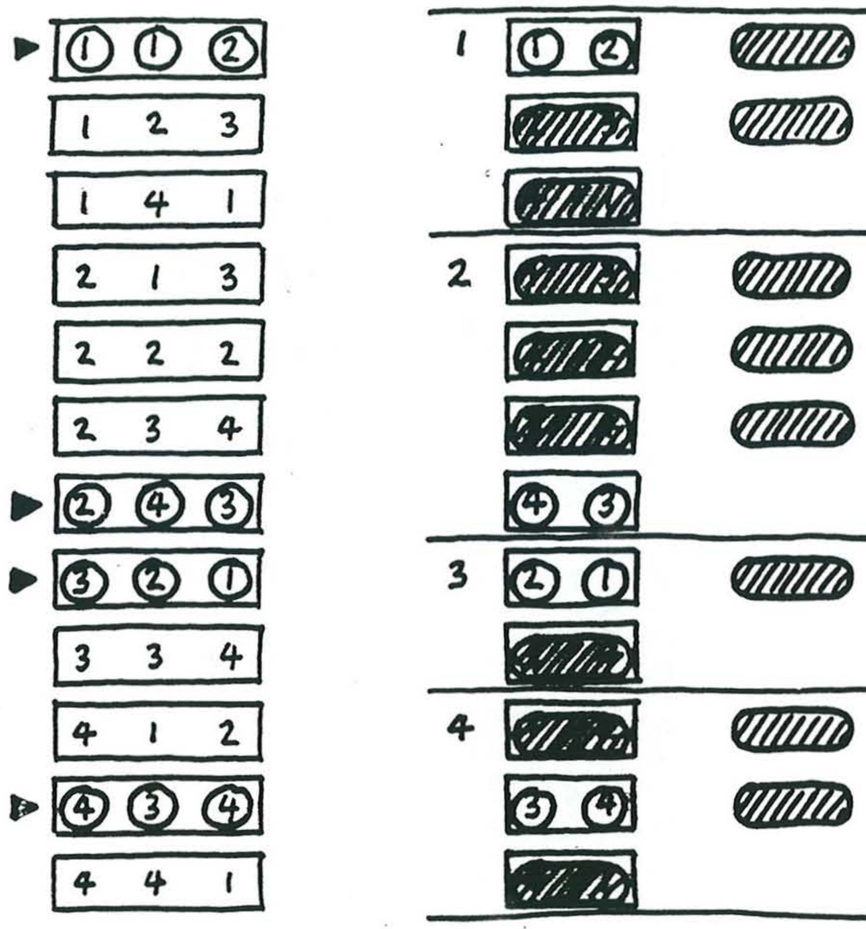
3DM $\propto$ $\leqslant 2$

3n ELEMENTS $\rightarrow$ 2n JOBS



$$p_{ij} = \begin{cases} 1 & \text{IF } j \in \text{TRIPLE } i \\ 3 & \text{OTHERWISE} \end{cases}$$

m TRIPLES $\rightarrow$ m MACHINES & m-n DUMMY JOBS



$$p_{ij} = \begin{cases} 2 & \text{FOR } i \text{ OF TYPE } 1 \\ 3 & \text{OTHERWISE} \end{cases}$$

$$p_{ij} = \begin{cases} 2 & \text{FOR } i \text{ OF TYPE } 2 \\ 3 & \text{OTHERWISE} \end{cases}$$

$$p_{ij} = \begin{cases} 2 & \text{FOR } i \text{ OF TYPE } 3 \\ 3 & \text{OTHERWISE} \end{cases}$$

$$p_{ij} = \begin{cases} 2 & \text{FOR } i \text{ OF TYPE } 4 \\ 3 & \text{OTHERWISE} \end{cases}$$

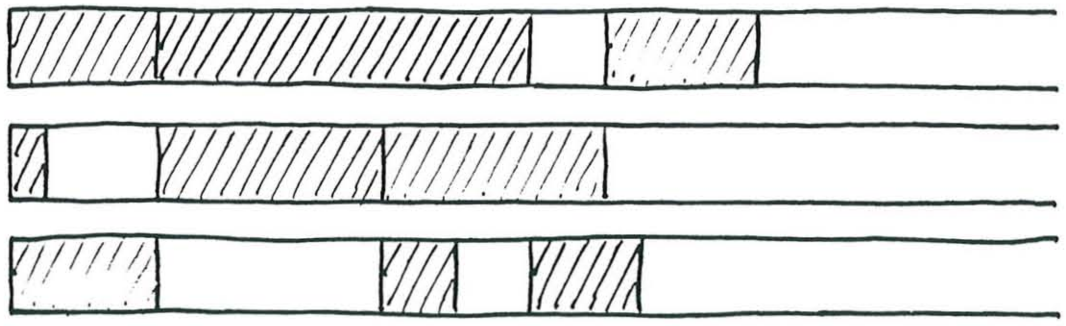
$\exists$ n TRIPLES $\Leftrightarrow$ $\exists$ SCHEDULE

CONTAINING OF LENGTH $\leqslant 2$

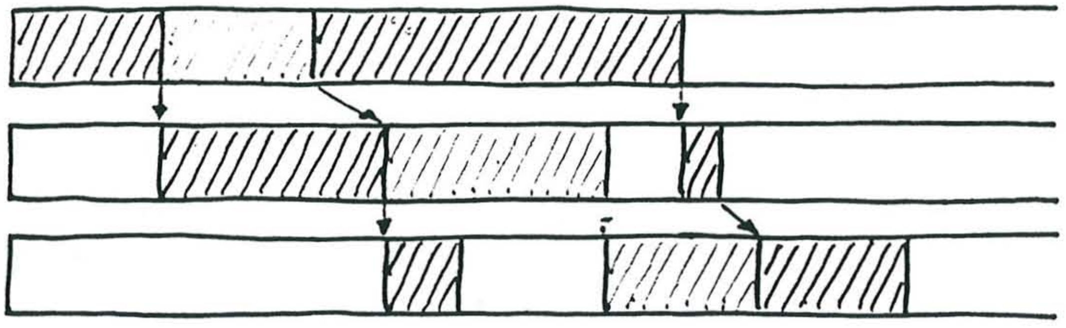
3n ELEMENTS

SHOP SCHEDULING

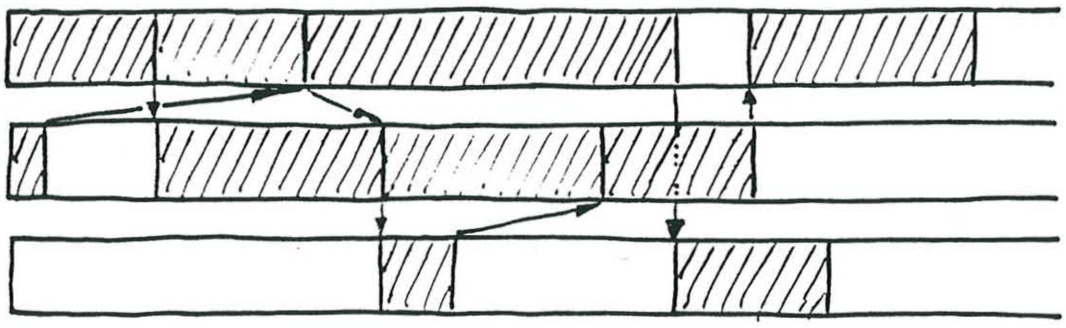
OPEN SHOP



FLOW SHOP



JOB SHOP



- INTEGRAL PROCESSING TIMES
- MINIMIZE SCHEDULE LENGTH

Oc		OPEN SHOP
Fc	GIVEN AN	FLOW SHOP INSTANCE & A $c \in \mathbb{N}$,
Jc		JOB SHOP

DOES THERE EXIST A SCHEDULE OF LENGTH $\leq c$?

* O2	* F2	* J2	[FOLKLORE]
* O3	* F3	* J3	[WILLIAMSON·H·H·H·L·S]
! O4	! F4	! J4	

IMPOSSIBILITIES (UNLESS $\mathcal{P}=\mathcal{NP}$):

- POLYNOMIAL ALGORITHM A WITH $R(A) < \frac{5}{4}$
- POLYNOMIAL APPROXIMATION SCHEME

POSSIBILITIES

O: $R(\text{GREEDY}) = 2$	[RACSMÁNY]
J: $R(\ldots) = O(\log^2(m \cdot m_{max}))$	[SHMOYS·STEIN·WEIN 1991]
F, m FIXED: PAS	[HALL 1995]
O, m FIXED: PAS	[SEVAST'JANOV· WOEGINGER 1996]

JOB SHOP SCHEDULING

JAN KAREL LENSTRA

EINDHOVEN UNIVERSITY OF TECHNOLOGY
CWI, AMSTERDAM

– LIMIT TO APPROXIMABILITY
Williamson · Hall · Hoogeveen · Hurkens · L · Sevast'janov · Shmoys

+ APPROXIMATION BY LOCAL SEARCH
Vaessens · Aarts · L

SET OF MACHINES

SET OF JOBS

EACH MACHINE • IS <u>AVAILABLE</u> AT TIME 0

• CAN HANDLE <u>$\leq$ 1 JOB</u> AT A TIME

EACH JOB • IS A <u>CHAIN</u> OF OPERATIONS

EACH OPERATION REQUIRES UNINTERRUPTED PROCESSING

ON GIVEN MACHINE

FOR GIVEN AMOUNT OF TIME

EXAMPLE: JOB A: $8 \rightarrow 3 \rightarrow 4$

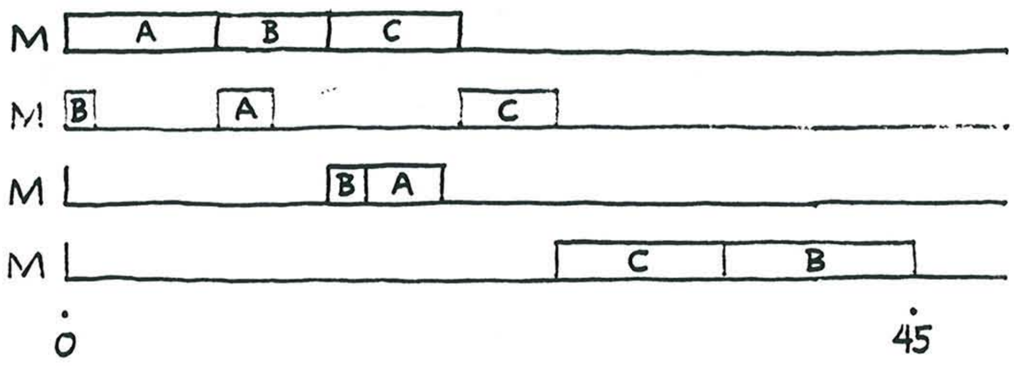
JOB B: $1 \rightarrow 6 \rightarrow 2 \rightarrow 10$

JOB C: $7 \rightarrow 5 \rightarrow 9$

SCHEDULE = ALLOCATION OF OPERATIONS

TO TIME INTERVALS ON MACHINES

EXAMPLE:



OPTIMAL SCHEDULE = SCHEDULE OF MINIMUM LENGTH

DISJUNCTIVE PROGRAM

SET $\mathcal{M}$ OF MACHINES

SET $\mathcal{J}$ OF JOBS

SET $\mathcal{O}$ OF OPERATIONS

OPERATION $i \in \mathcal{O}$
- BELONGS TO JOB $J_i \in \mathcal{J}$
- REQUIRES PROCESSING ON MACHINE $M_i \in \mathcal{M}$
- DURING TIME $p_i \in \mathbb{N}$

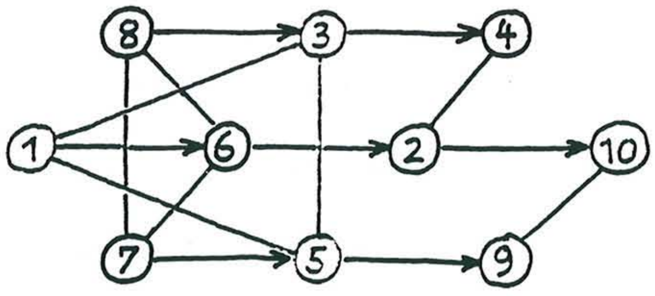
BINARY RELATION $\rightarrow$ ON $\mathcal{O}$
- DECOMPOSING $\mathcal{O}$ INTO CHAINS (= JOBS)

FIND STARTING TIMES S_i $(i \in \mathcal{O})$
- MINIMIZING
  - LENGTH: $\max_{i \in \mathcal{O}} \{S_i + p_i\}$
- SUBJECT TO
  - AVAILABILITY: $S_i \geqslant 0$ $\quad (i \in \mathcal{O})$
  - PRECEDENCE: $S_j - S_i \geqslant p_i$ $\quad (i,j \in \mathcal{O},\ i \rightarrow j)$
  - CAPACITY: $S_j - S_i \geqslant p_i \ \vee\ S_i - S_j \geqslant p_j$ $\quad (i,j \in \mathcal{O},\ M_i = M_j)$

DISJUNCTIVE GRAPH $G = (\mathcal{O}, A, E)$

- VERTEX SET $\mathcal{O}$
- ARC SET $\quad A = \{(i,j) \mid i \longrightarrow j\}$ ... DIRECTED!
- EDGE SET $\quad E = \{\{i,j\} \mid M_i = M_j\}$ ... UNDIRECTED!
- WEIGHT $\quad p_i$ FOR EACH $i \in \mathcal{O}$

EXAMPLE:



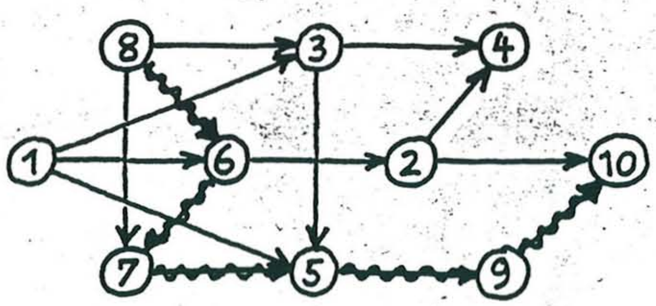
CAPACITY CONFLICT: EDGE $\{i,j\} \in E$

SCHEDULING DECISION: REPLACE $\{i,j\}$ BY (i,j) OR (j,i)

DISJUNCTIVE GRAPH $G = (\mathcal{O}, A, E)$

- VERTEX SET $\mathcal{O}$
- ARC SET $A = \{(i,j) \mid i \rightarrow j\}$ ... DIRECTED!
- EDGE SET $E = \{\{i,j\} \mid M_i = M_j\}$ ... UNDIRECTED!
- WEIGHT p_i FOR EACH $i \in \mathcal{O}$

EXAMPLE:



CAPACITY CONFLICT: EDGE $\{i,j\} \in E$

SCHEDULING DECISION: REPLACE $\{i,j\}$ BY (i,j) OR (j,i)

FEASIBLE SCHEDULE: ORIENTATION $\bar{E}$ OF E SUCH THAT DIGRAPH $\bar{G} = (\mathcal{O}, A \cup \bar{E})$ IS ACYCLIC

LENGTH OF SCHEDULE: LONGEST PATH LENGTH IN $\bar{G}$

PROBLEM: FIND ORIENTATION $\bar{E}$ OF E THAT MINIMIZES LONGEST PATH LENGTH IN $\bar{G}$

	POLYNOMIAL ALGORITHMS	EXPONENTIAL ALGORITHMS
OPTIMAL SOLUTIONS	FAST OPTIMIZATION	ENUMERATION: DP, B&B
APPROXIMATE SOLUTIONS	FAST APPROXIMATION	LOCAL SEARCH

COMPUTATIONAL COMPLEXITY

POLYNOMIAL TIME	NP-HARD
2 MACHINES ≤ 2 OPERATIONS/JOB	2 MACHINES ≤ 3 OPERATIONS/JOB 3 MACHINES ≤ 2 OPERATIONS/JOB
2 MACHINES ALL $p_i = 1$	2 MACHINES ALL $p_i \in \{1, 2\}$ 3 MACHINES ALL $p_i = 1$
2 JOBS	3 JOBS
LENGTH ≤ 3	LENGTH ≤ 4

$J4$ FIND SCHEDULE OF LENGTH ≤ 4 NP-HARD

$\Downarrow$

$J\frac{5}{4}$ FIND SCHEDULE OF LENGTH $< \frac{5}{4} \times$ OPTIMUM NP-HARD

$J4 \propto J\frac{5}{4}$:

SUPPOSE A IS POLYNOMIAL ALGORITHM FOR $J\frac{5}{4}$

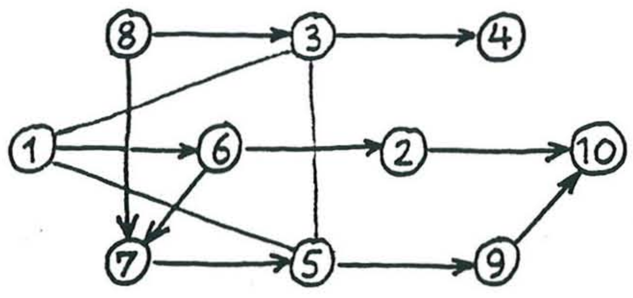
TAKE ANY INSTANCE I

$$\left.\begin{array}{ll} \bullet\ OPT(I) \geqslant 5 \Rightarrow & A(I) \geqslant 5 \\ \bullet\ OPT(I) \leqslant 4 \Rightarrow A(I) < \frac{5}{4} \cdot OPT(I) \leqslant 5 \Rightarrow & A(I) \leqslant 4 \end{array}\right\}$$

$\Longrightarrow$ A IS POLYNOMIAL ALGORITHM FOR $J4$

BRANCH & BOUND

NODE: SOME $E' \subset E$ HAS ORIENTATION $\overline{E'}$



LOWER BOUNDS

LB0
- IGNORE EDGES IN $E - E'$
- COMPUTE LONGEST PATH LENGTH IN $(\mathcal{O}, A \cup \overline{E'})$

LB1
- CHOOSE M^*
- IGNORE EDGES IN $E - E'$ <u>NOT</u> ON M^*
- SOLVE 1-MACHINE PROBLEM WITH FOR EACH i ON M^*:
  - HEAD r_i = LONGEST PATH LENGTH UP TO i
  - BODY p_i = PROCESSING TIME
  - TAIL q_i = LONGEST PATH LENGTH FROM i

STRENGTHEN LB1
- PRECEDENCE CONSTRAINTS: $\textcircled{1} \longrightarrow \textcircled{5}$
- PRECEDENCE DELAYS: $\textcircled{1} \xrightarrow{13} \textcircled{5}$
- ADJUSTED HEADS & TAILS: $r_j = \max(8+7, 1+6+7) \rightarrow r_j = 8+6$
- LBk, $k > 1$

SURROGATE DUALITY BOUNDS

POLYHEDRAL BOUNDS

LOWER BOUND VALUES

FOR 10X10 INSTANCE [Fischer, Thompson, 1963]

OPTIMUM	930		[Carlier, Pinson, 1989]
LB1	808		[McMahon, Florian, 1975]
LB1 WITH ADJUSTED HEADS & TAILS	868		[Carlier, Pinson, 1994]
LB5	907		[Lageweg, 1984]
SURROGATE DUALITY	813	>1 HR	[Fisher, Lageweg, L, Rinnooy Kan, 1983
POLYHEDRAL BOUNDS			[Applegate, Cook, 1991]
• CUTS 1	823	5 SEC	
• CUTS 2	824	5 MIN	
• CUTS 3	827	>2 HR	

BRANCHING RULES

(a) GENERATE 'ACTIVE SCHEDULES'

(b) ORIENT CRUCIAL EDGE

(c) APPLY 'BLOCK APPROACH'

IMPLEMENTATIONS

- UB: APPROXIMATION ALGORITHM, E.G, SHIFTING BOTTLENECK
- LB: PREEMPTIVE LB1
- BRANCHING: (b) OR (c)
- ELIMINATION CRITERIA: MANY!

RESULTS FOR 10x10 INSTANCE

- 22021 NODES, 300 MIN [Carlier, Pinson, 1989]
- 16055 NODES, 6 MIN [Applegate, Cook, 1991]
- 4242 NODES, 19 MIN [Brucker, Jurisch, Sievers, 1992]
- 37 NODES, 8 MIN [Carlier, Pinson, 1994]

MAJOR ISSUE

- FIND BETTER (LP-BASED?) LOWER BOUNDS

APPROXIMATION ALGORITHMS: CONSTRUCTION

PRIORITY RULE

- SCHEDULE OPERATIONS ACCORDING TO SOME PRIORITY FUNCTIC

ACTIVE SCHEDULES

<u>DEFINITION</u>: A SCHEDULE IS 'ACTIVE'

IF MOVING BACK ONE OPERATION WILL DELAY ANOTHER ONE.

<u>THEOREM</u>: AT LEAST ONE OPTIMAL SCHEDULE IS ACTIVE.

<u>CONSTRUCTION</u>:

- LET S_i = EARLIEST POSSIBLE STARTING TIME OF i ($i \in \mathcal{O}$)

  LET $\mathcal{O}'$ = SET OF UNSCHEDULED OPERATIONS
- DETERMINE $j \in \mathcal{O}'$ SUCH THAT

  $$S_j + p_j = \min_{i \in \mathcal{O}'} \{S_i + p_i\}$$
- SELECT OPERATION FROM SET

  $$\{i \mid i \in \mathcal{O}', M_i = M_j, \; S_i < S_j + p_j\}$$

  ACCORDING TO SOME PRIORITY FUNCTION

LOCAL SEARCH

F	:	SET OF FEASIBLE SOLUTIONS x
$c(x) \in \mathbb{R}$	:	COST OF x
$N(x) \subset F$	:	NEIGHBORHOOD OF x

ITERATIVE IMPROVEMENT

- GENERATE $x \in F$
- AS LONG AS $\exists\, y \in N(x)$ WITH $c(y) < c(x)$, DO $x \leftarrow y$
- x IS <u>LOCALLY OPTIMAL</u> WITH RESPECT TO N

NEW VARIANTS

- SIMULATED ANNEALING
- THRESHOLD ACCEPTANCE
- TABU SEARCH
- VARIABLE-DEPTH SEARCH
- GENETIC ALGORITHMS
- ...

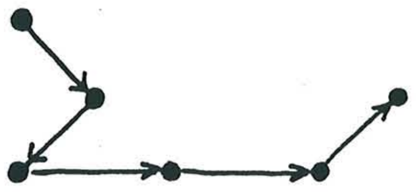
HYBRID VARIETIES

LOCAL SEARCH COMBINED WITH

- CONSTRUCTIVE RULE
- OTHER LOCAL SEARCH METHOD
- BACKTRACKING SCHEME

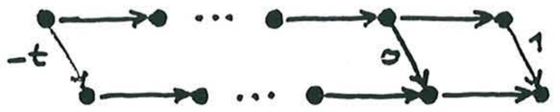
SMALL-CHANGE NEIGHBORHOODS

S SWAP: REVERSE MACHINE ARC ON LONGEST PATH



- SWAP ELSEWHERE WILL NOT GIVE IMPROVEMENT
- SWAP ON LONGEST PATH WILL NOT CREATE CYCLE
- ∀SCHEDULE ∃ PATH TO OPTIMAL SCHEDULE

S+ SWAP CRITICAL ARC 0 AND ARCS −t & 1



J JUMP: REINSERT OPERATION ON LONGEST PATH ELSEWHERE

P PERMUTE 3 OR MORE OPERATIONS ON LONGEST PATH

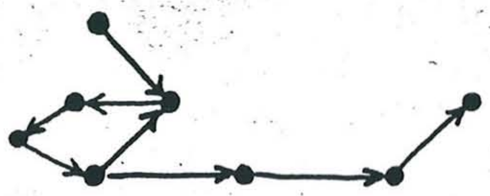
BIG-CHANGE NEIGHBORHOODS

R1 RESCHEDULE 1 MACHINE ··· BY COMPUTING LB1

Rt RESCHEDULE t MACHINES ··· BY JOB SHOP SCHEDULING

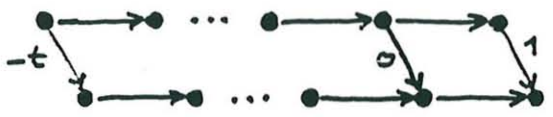
SMALL-CHANGE NEIGHBORHOODS

S SWAP: REVERSE MACHINE ARC ON LONGEST PATH



- SWAP ELSEWHERE WILL NOT GIVE IMPROVEMENT
- SWAP ON LONGEST PATH WILL NOT CREATE CYCLE
- ∀SCHEDULE ∃ PATH TO OPTIMAL SCHEDULE

S+ SWAP CRITICAL ARC 0 AND ARCS −t & 1



J JUMP: REINSERT OPERATION ON LONGEST PATH ELSEWHERE

P PERMUTE 3 OR MORE OPERATIONS ON LONGEST PATH

BIG-CHANGE NEIGHBORHOODS

R1 RESCHEDULE 1 MACHINE ... BY COMPUTING LB1

Rt RESCHEDULE t MACHINES ... BY JOB SHOP SCHEDULING

SHIFTING BOTTLENECK [Adams, Balas, Zawack, 1988]

FOR k ← 1 TO m DO:

- CONSTRUCTION:

  SCHEDULE MACHINE MAXIMIZING LB1 AMONG UNSCHEDULED MACHINES

- ITERATIVE IMPROVEMENT USING R1:

  REOPTIMIZE PARTIAL SCHEDULE BY RESCHEDULING 1 MACHINE AT A TIME

VARIATIONS

- SB + DELAYS [Balas, L, Vazacopoulos, 1995]

  SCHEDULE MACHINE MAXIMIZING LB1 INCORPORATING PRECEDENCE DELAYS

- SB + GUIDED LOCAL SEARCH [Balas, Vazacopoulos, 1994]

  REOPTIMIZE BY VARIABLE-DEPTH SEARCH USING JUMPS

HYBRIDS

- PARTIAL ENUMERATION [Adams, Balas, Zawack, 1988]

  SUB-BRANCH & SUPER-BOUND, APPLYING SB IN EACH NODE

- BOTTLE-t [Applegate, Cook, 1991]

  FOR k > m-t: BRANCH, CONSIDERING EACH MACHINE IN TURN

- SHUFFLE [Applegate, Cook, 1991]
  - CONSTRUCT SCHEDULE USING BOTTLE-5
  - ITERATIVELY IMPROVE IT USING Rt

RESULTS

SB >> {SB·DELAYS, PE, BOTTLE-4,5,6} > SHUFFLE >> SB·GLS

II ITERATIVE IMPROVEMENT [Aarts, vLaarhoven, L, Ulder, 1994]

ACCEPT y IF $c(y)-c(x) < 0$

TA THRESHOLD ACCEPTING [Aarts, vLaarhoven, L, Ulder, 1994]

ACCEPT y IF $c(y)-c(x) < t$ WITH $t \downarrow 0$

SA SIMULATED ANNEALING [vLaarhoven, Aarts, L, 1992]

ACCEPT y IF $c(y)-c(x) < 0$ OR WITH PROBABILITY $\downarrow 0$

SA!! BI-LEVEL VARIANT [Matsuo, Suh, Sullivan, 1988]

- ACCEPT y AS IN SA
- OTHERWISE, SUBJECT y TO !! TO OBTAIN z

  ACCEPT z AS IN !!

RESULTS

- TIME EQUIVALENT

  $!!_S > !!_{S+} \gg TA_S \gg \{PE, SA_S, SA_{S+}\} > SHUFFLE > SA!!_{S+}$

- TIME OF NO CONCERN

  $SHUFFLE \gg SA_S(\infty)$

CONCLUSIONS

- RANDOMIZATION HELPS
- EXPLOITING PROBLEM STRUCTURE HELPS

TABOO SEARCH

- SELECT BEST NEIGHBOR

  (UNLESS IT IS FORBIDDEN (UNLESS IT IS GOOD ENOUGH))
- MAINTAIN FORBIDDEN SET

  (OFTEN DEFINED IN TERMS OF FORBIDDEN MOVES)

TS_S	SWAP CRITICAL ARC	[Barnes, Chambers, 1995]
TS_P	PERMUTE MORE CRITICAL OPERATIONS	[Dell'Amico, Trubian, 1993]
$TS_S BT$	BACKTRACKING TO REJECTED MOVES	[Nowicki, Smutnicki, 1995]

$TS_S > SA_S(\infty) > TS_P > TS_S BT$

VARIABLE-DEPTH SEARCH

- $x \rightarrow$ CHAIN OF SOLUTIONS, MAKING SMALL GREEDY MOVES
- $y \leftarrow$ BEST SOLUTION IN CHAIN

GLS_J GUIDED LOCAL SEARCH [Balas, Vazacopoulos, 1994]

- TREES INSTEAD OF CHAINS, BOUNDED BY HEURISTIC RULES
- MOVES ARE JUMPS

IGLS ITERATED GLS

SB·GLS $\rightarrow$ [GLS ON $m-1$ MACHINES $\rightarrow$
GLS ON m MACHINES $]^{\infty}$

RGLS-k REITERATED GLS

IGLS $\rightarrow$ [GLS ON $m-\sqrt{m}$ MACHINES $\rightarrow$
SB TO ADD $\sqrt{m}$ MACHINES $\rightarrow$
IGLS ON m MACHINES $]^{k}$

GLS > IGLS > $TS_S BT$ > RGLS-5 ; $TS_S BT$ IS FASTER

GENETIC ALGORITHMS

- CHOOSE POPULATION OF SOLUTIONS, SPLIT IN PAIRS
- FOR EACH PAIR, GENERATE TWO HYPERNEIGHBORS
- REDUCE POPULATION TO ORIGINAL SIZE

VARIETY OF HYPERNEIGHBORHOODS
ALLOWING USE OF

- STRING REPRESENTATIONS
- MUTATIONS
- CROSSOVERS

BILEVEL VARIANTS
SUBJECTING HYPERNEIGHBORS TO, E.G., II

RESULTS

- OFTEN POOR
- AT BEST IN RANGE [SB, SHUFFLE]

CONSTRAINT SATISFACTION

SOLVE FEASIBILITY PROBLEM BY TREE SEARCH USING

- [BRANCH] VARIABLE AND VALUE SELECTION
- [BOUND] CONSISTENCY ENFORCING

← COMPLETE ENUMERATION TRADITION OF LOGIC PROGRAMMING

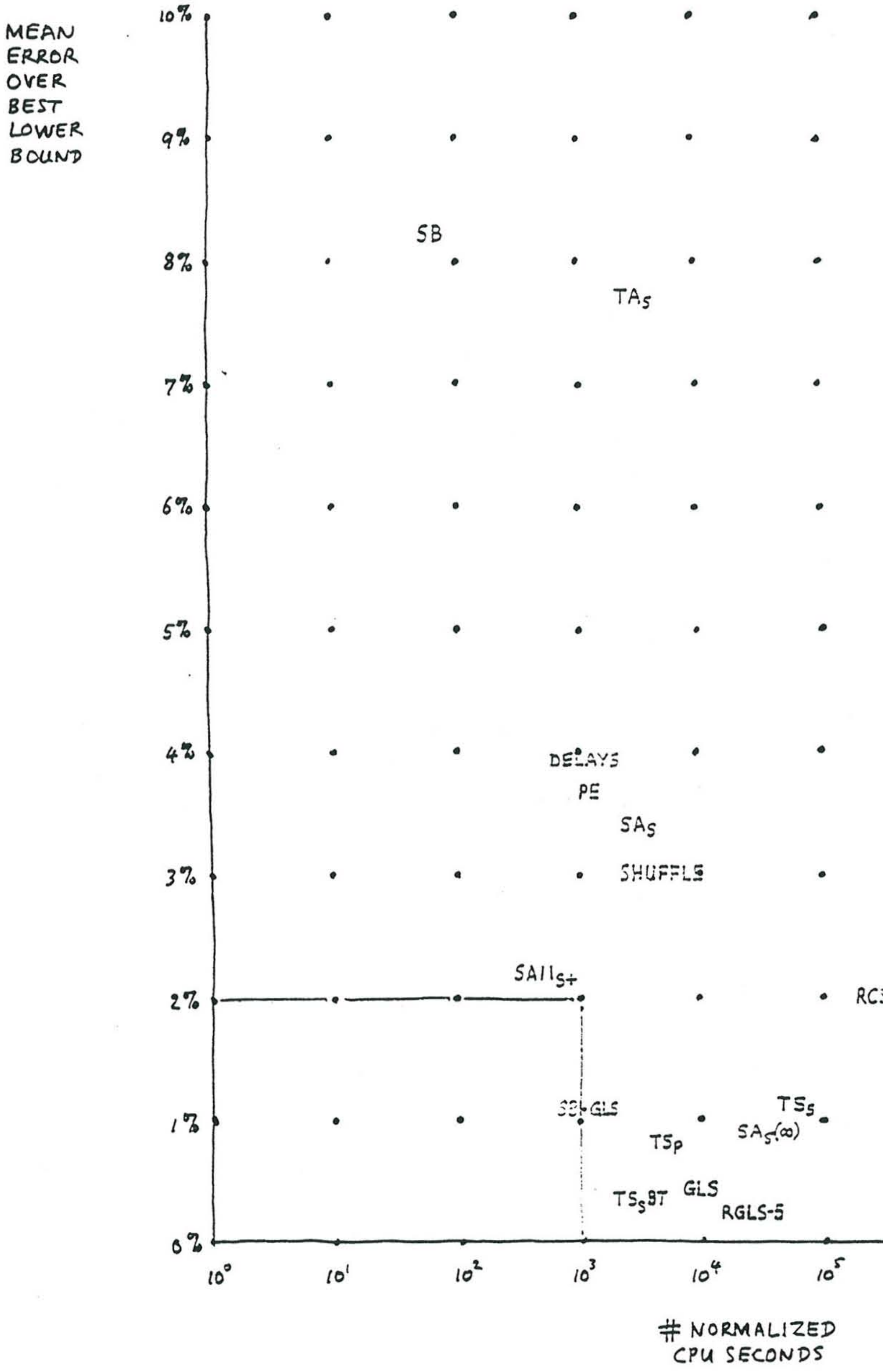
→ COMBINATION WITH TECHNIQUES FROM MATH. PROGRAMMING

RESULTS

RANDOMIZED RCS IS OK BUT SLOW

NEURAL NETWORKS

∅


MEAN ERROR OVER BEST LOWER BOUND
10%
9%
8%
7%
6%
5%
4%
3%
2%
1%
0%
SB
TA$_S$
DELAYS
PE
SA$_S$
SHUFFLE
SAII$_{S+}$
RC
SB+GLS
TS$_S$
TS$_P$
SA$_S$(∞)
TS$_S$BT
GLS
RGLS-5
10^0
10^1
10^2
10^3
10^4
10^5
# NORMALIZED CPU SECONDS

DISCUSSION

Rapporteur: Martin Beet

Lecture One

Referring to the TSP problem mentioned in the talk, dr Arne Andersson inquired about the alleged success of the neural network approach compared to the use of linear programming or other techniques. Professor Lenstra was somewhat sceptical about this, and replied that in his opinion this kind of combinatorial problem did not lend itself to effective solution by neural nets. In his view many researchers were uncertain about the great power and wide-ranging applicability of linear programming.

Professor Nievergelt inquired about scheduling research concerning the problem of disturbed schedules, as might be encountered in transport management; for example how to handle a delayed train or airline flight while causing as little disturbance to the remaining schedule as possible. Professor Lenstra stated that this robustness of a schedule seemed to be highly dependent on the initial schedule, but that most researchers in scheduling had concentrated on static problems. Mr Ainsworth supported this view. Both agreed on the importance and the demand for techniques to solve these problems.

Lecture Two

In answering a query from the audience, Professor Lenstra stated that the comparison of the running time of different local-search algorithms relied solely on empirical findings, as a unifying theory was not available.

Professor Mehlhorn was curious as to why the linear programming approach was not also used for job shop scheduling (JSS). Professor Lenstra explained that in his experience the LP approach with relaxations did not provide satisfactory results, due to the difficulty of dealing with the disjunction in the formulation of the precedence constraints. He added that local-search algorithms were a valid approach, since there seemed to be little demand for algorithms to solve large-scale JSS problems.