Job Shop Scheduling Problem – Population Based Metaheuristics

LUCAS PIEREZAN MAGALHÃES

Summary

- Job Shop Scheduling Problem (JSSP)
  - Definition
  - Disjunctive Graph Representation (DG)
  - Semi-Active/Active Scheduling
- Literature Review
  - Instances and Landmarks
  - The BRKGA Approach
  - The Path Relink + Tabu Search Approach
- Our Work
  - Proposed Method
  - Results

The JJSP Problem

Schedule set of Jobs $J = \{J_1, J_2, \dots, J_n\}$ using Machines $M = \{M_1, M_2, \dots, M_m\}$.

Each job j has m operations $O_j = \{o_{j,1}, o_{j,2}, .., o_{j,m}\}$ that must be executed in order.

Each operation $o_{j,i}$ has to executed in a specific machine and takes a certain process time.

TABLE 1. Problem data for 4-job, 3-machine example.

Seq. Order	J_1		J_2		J_3		J_4	
	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time
1	a	2	b	3	c	5	b	2
2	b	3	c	2	b	2	a	4
3	c	4	a	3	a	3	c	2

[BRKGA]

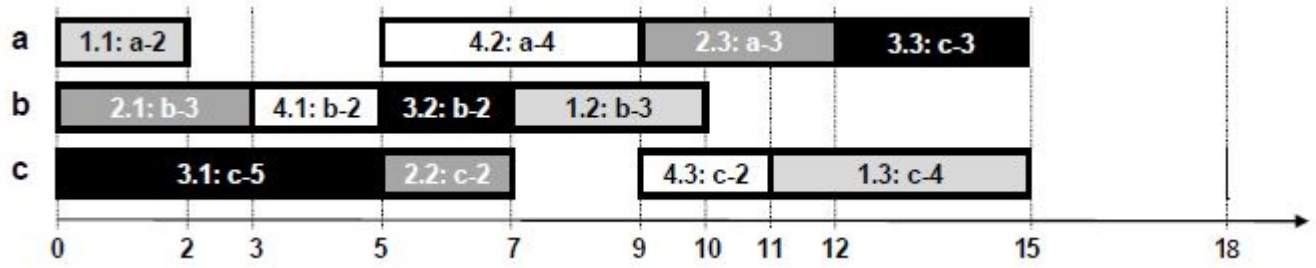
The JJSP Problem

Ex.

TABLE 1. Problem data for 4-job, 3-machine example.

Seq. Order	J_1		J_2		J_3		J_4	
	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time
1	a	2	b	3	c	5	b	2
2	b	3	c	2	b	2	a	4
3	c	4	a	3	a	3	c	2

A possible solution:



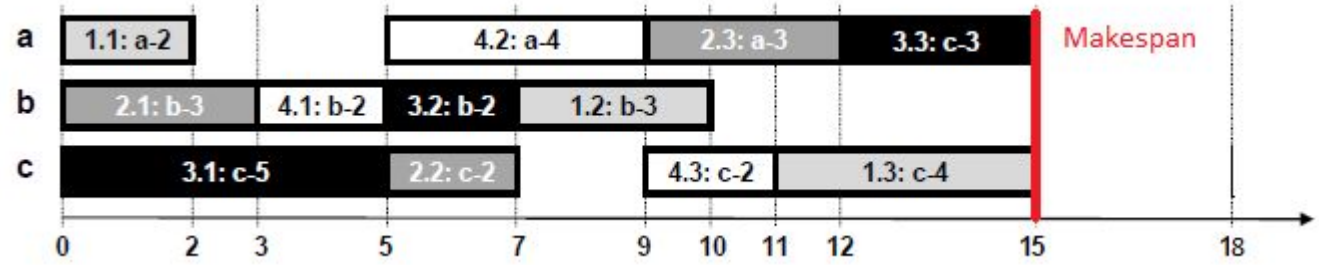
The JJSP Problem

Objective: Makespan minimization.

Restrictions:

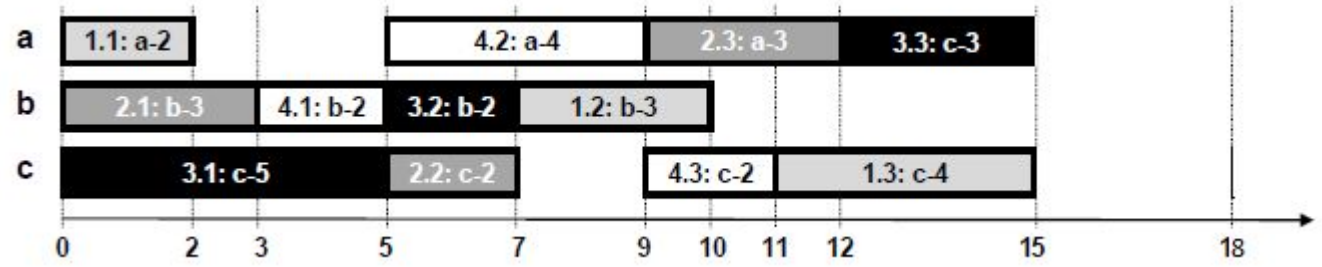
Job operations order.

Each machine only process one operation at time.



Basic Concepts

A solution can be represented by the execution order of jobs on each machine:



Machines Permutations of Jobs

Machine				
a	J1	J4	J2	J3
b	J2	J4	J3	J1
c	J3	J2	J4	J1

Basic Concepts

Not every such m permutations of n elements is a feasible solution. **Deadlock!**

TABLE 1. Problem data for 4-job, 3-machine example.

	J_1		J_2		J_3		J_4	
Seq. Order	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time
1	a	2	b	3	c	5	b	2
2	b	3	c	2	b	2	a	4
3	c	4	a	3	a	3	c	2

Machine			
a	J4	..	J1
b	J1	..	J4
c	..	..	..

Basic Concepts

Disjunctive Graph Model(DG Model):

Introducing job precedence.

TABLE 1. Problem data for 4-job, 3-machine example.

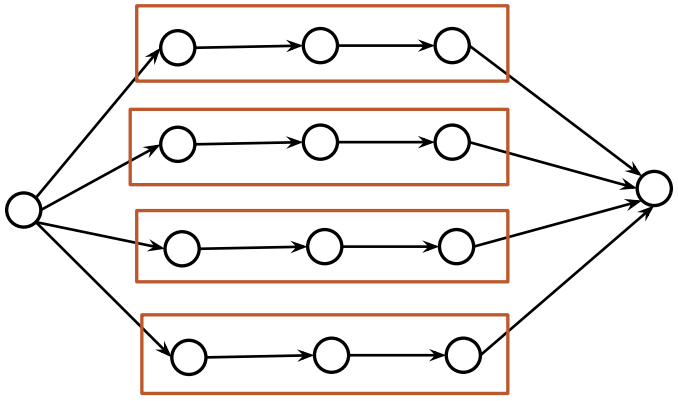
Seq. Order	J_1		J_2		J_3		J_4	
	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time
1	a	2	b	3	c	5	b	2
2	b	3	c	2	b	2	a	4
3	c	4	a	3	a	3	c	2

J1

J2

J3

J4



Job j operations in order

Basic Concepts

Disjunctive Graph Model(DG Model):

Same machine operations.

Machines:

a

b

c

TABLE 1. Problem data for 4-job, 3-machine example.

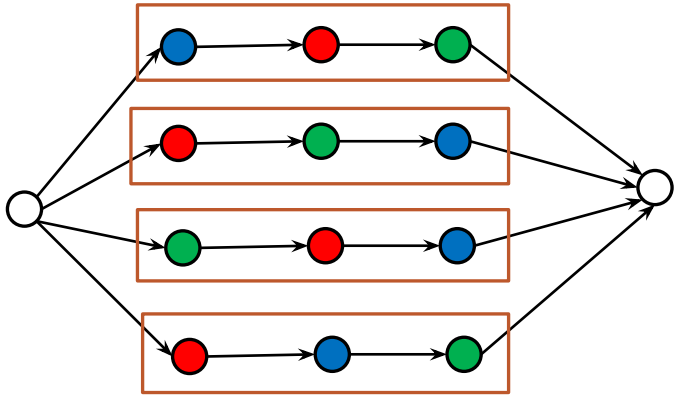
	J_1		J_2		J_3		J_4	
Seq. Order	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time
1	a	2	b	3	c	5	b	2
2	b	3	c	2	b	2	a	4
3	c	4	a	3	a	3	c	2

J1

J2

J3

J4



Basic Concepts

Disjunctive Graph Model(DG Model):

A solution is a set of "machine precedente edges" that makes the graph acyclic.

Machines:

a

b

c

TABLE 1. Problem data for 4-job, 3-machine example.

	J_1		J_2		J_3		J_4	
Seq. Order	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time
1	a	2	b	3	c	5	b	2
2	b	3	c	2	b	2	a	4
3	c	4	a	3	a	3	c	2

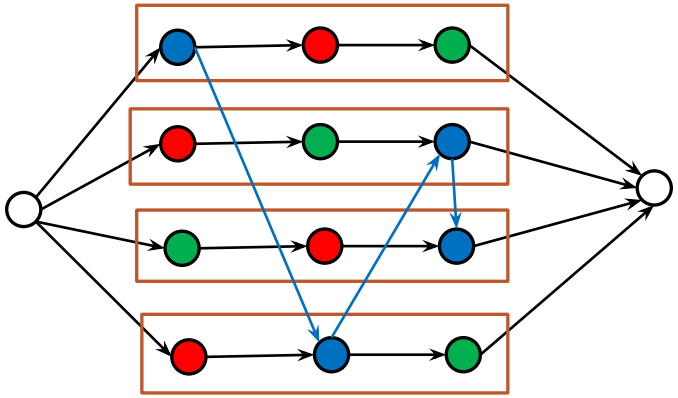
only machine a

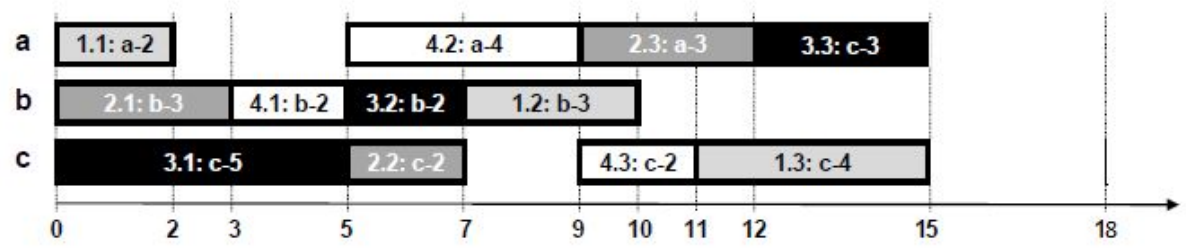
J1

J2

J3

J4





Basic Concepts

Disjunctive Graph Model(DG Model):

A solution is a set of “machine precedente edges” that makes the graph acyclic.

Machines:

a

b

c

TABLE 1. Problem data for 4-job, 3-machine example.

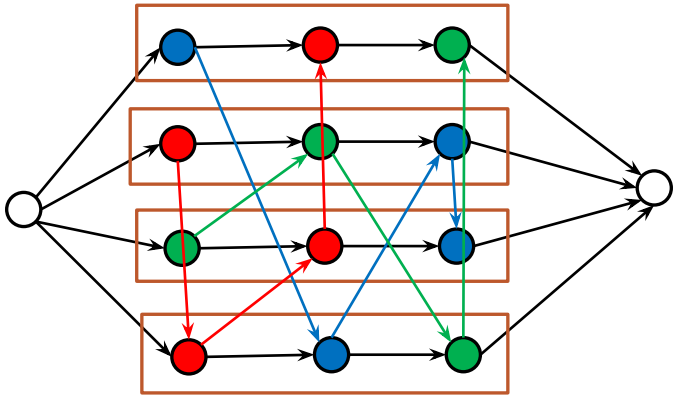
	J_1		J_2		J_3		J_4	
Seq. Order	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time
1	a	2	b	3	c	5	b	2
2	b	3	c	2	b	2	a	4
3	c	4	a	3	a	3	c	2

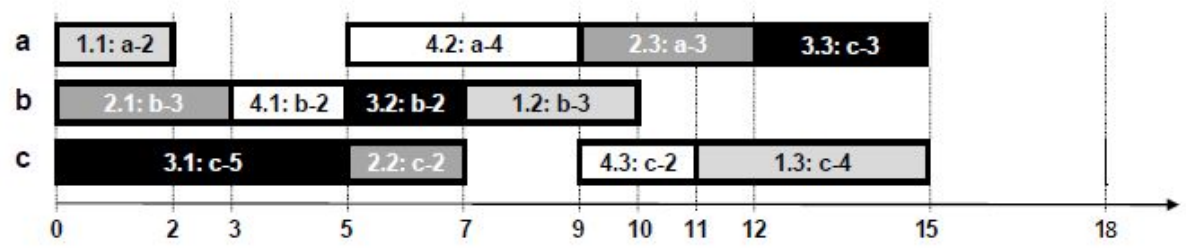
J1

J2

J3

J4





Basic Concepts

Disjunctive Graph Model(DG Model):

Proc. time in vertex (or edges) cost.

Any longest path in this graph is called a critical path.

Machines:

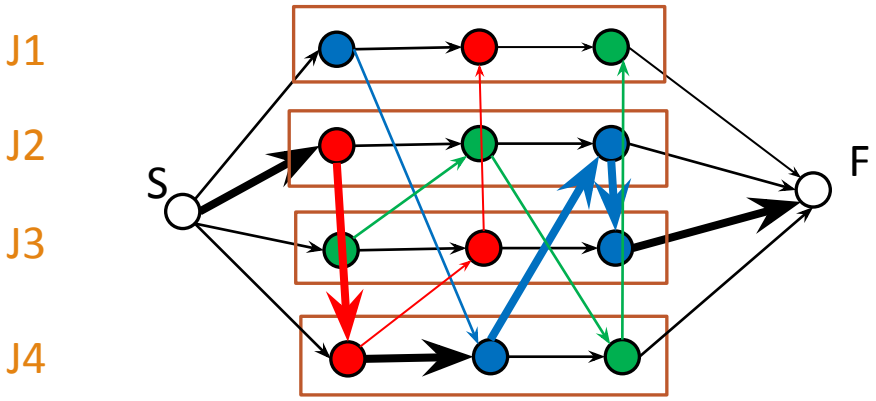
a

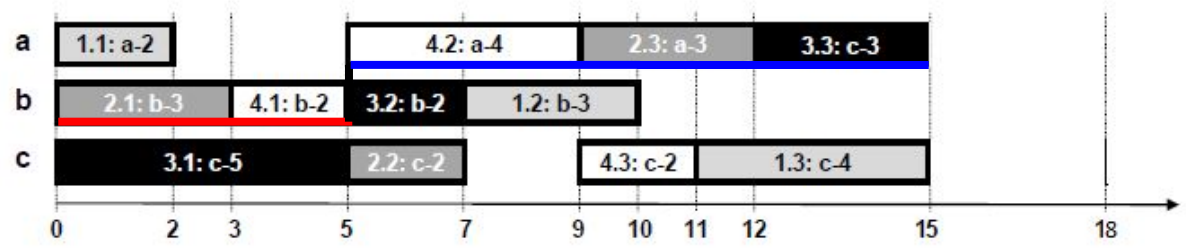
b

c

TABLE 1. Problem data for 4-job, 3-machine example.

	J_1		J_2		J_3		J_4	
Seq. Order	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time
1	a	2	b	3	c	5	b	2
2	b	3	c	2	b	2	a	4
3	c	4	a	3	a	3	c	2





Basic Concepts

Disjunctive Graph Model(DG Model):

Proc. time in vertex (or edges) cost.

The makespan of a solution is the longest path from S to F.

Machines:

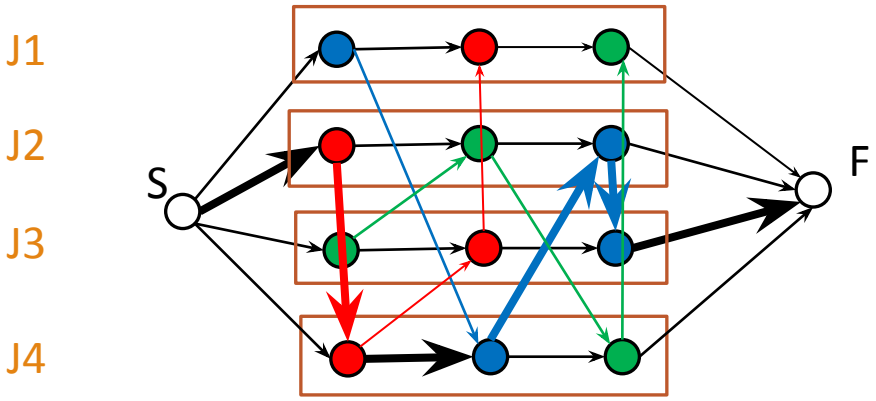
a

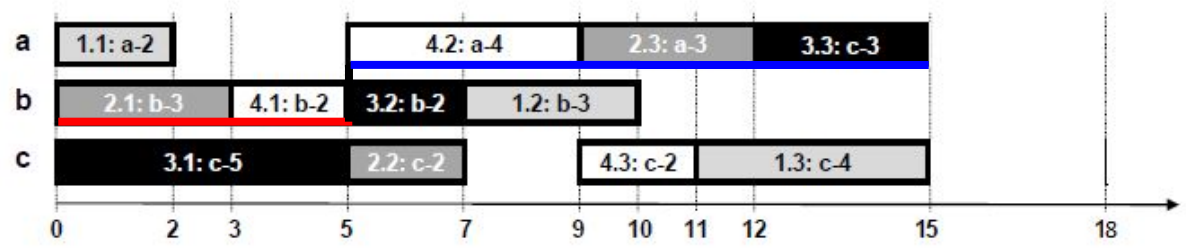
b

c

TABLE 1. Problem data for 4-job, 3-machine example.

	J_1		J_2		J_3		J_4	
Seq. Order	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time
1	a	2	b	3	c	5	b	2
2	b	3	c	2	b	2	a	4
3	c	4	a	3	a	3	c	2





Basic Concepts

Disjunctive Graph Model(DG Model):

The JSSP is to find such acyclic orientation with minimum longest path.

Machines:

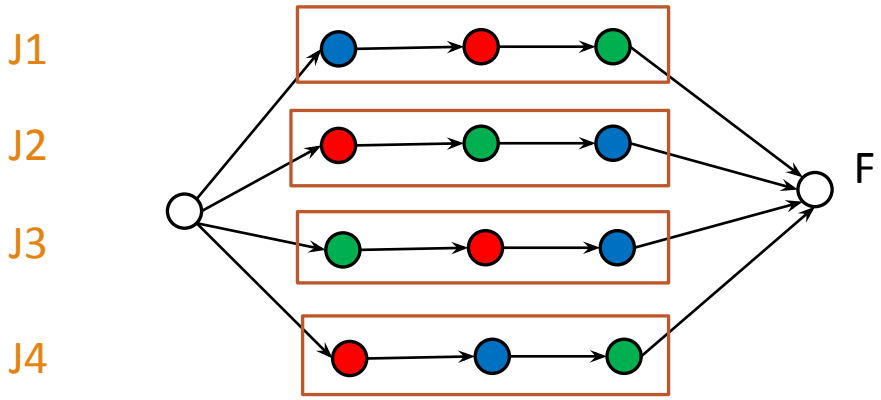
a

b

c

TABLE 1. Problem data for 4-job, 3-machine example.

	J_1		J_2		J_3		J_4	
Seq. Order	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time	Mach.	Proc. time
1	a	2	b	3	c	5	b	2
2	b	3	c	2	b	2	a	4
3	c	4	a	3	a	3	c	2



Literature Review

- Early approaches
  - Using constructive heuristics with “priority rules” (Active Schedules)
  - Population methods:
    - Binary representation (edges orientation)
    - Permutations (m permutations or 1 permutation with repetition)

Literature Review

Recent work considered in the project:

- Nowicki and Smutnicki (1996, 2005)[4,5]: Tabu Search, N5 Neighborhood, Path-relink diversification.
- Balas and Vazacopoulos (1998)[2] : N6 Neighborhood.
- Zhang et al. (2007)[6]: N7 Neighborhood + Tabu Search
- Gonçalves and Resende (2013)[3]: BRKGA, Akers extended local search.
- Bo Peng, Lu, Cheng(2014)[1]: Tabu Search and Path Relink.

Key Ideas:

- Critical path destruction Neighborhoods (N5,N6,N7) and efficient exploration.
- Combination and Repair of solutions.

Proposed Method

The proposed method combines:

Population Policy from BRKGA:

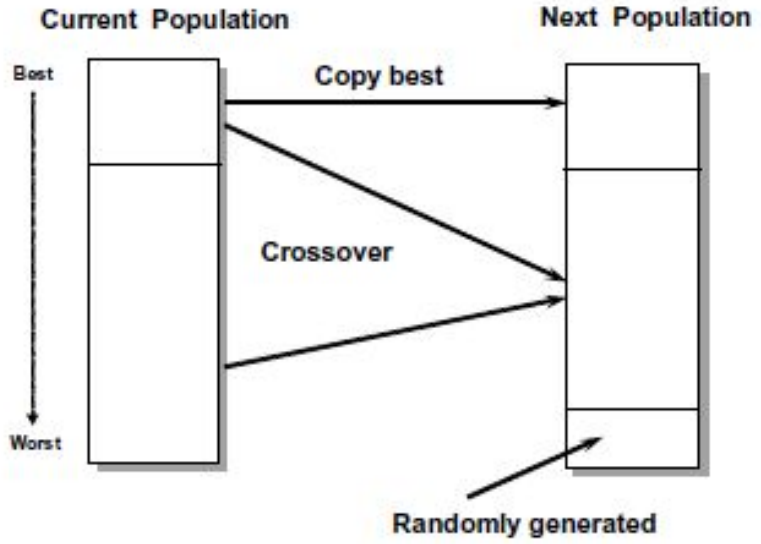
A Biased Random-Key Genetic Algorithm for Job-Shop Scheduling (BRKGA),
Gonçalves and Resende (2013)[3]

With Crossover from PR/TS [1]:

A Tabu Search/Path Relinking Algorithm to Solve the Job Shop Scheduling Problem
Bo Peng, Zhipeng Lu, T.C.E. Cheng.

Proposed Method

```
P = Initial_Population()
Tabu_Search(P)
While Stop_Criteria_Test()
        Offspring = Generate_Offspring(P)
        Offspring = Tabu_Search(Offspring)
        Mutants = Generate_Random_Solutions()
        Mutants = Tabu_Search(Mutants)
        Population = Elite(Population) + Offspring + Mutants
```



Proposed Method

Representation is interchanged when necessary but we use manly,

Permutations of operations for each machine.

Ex.

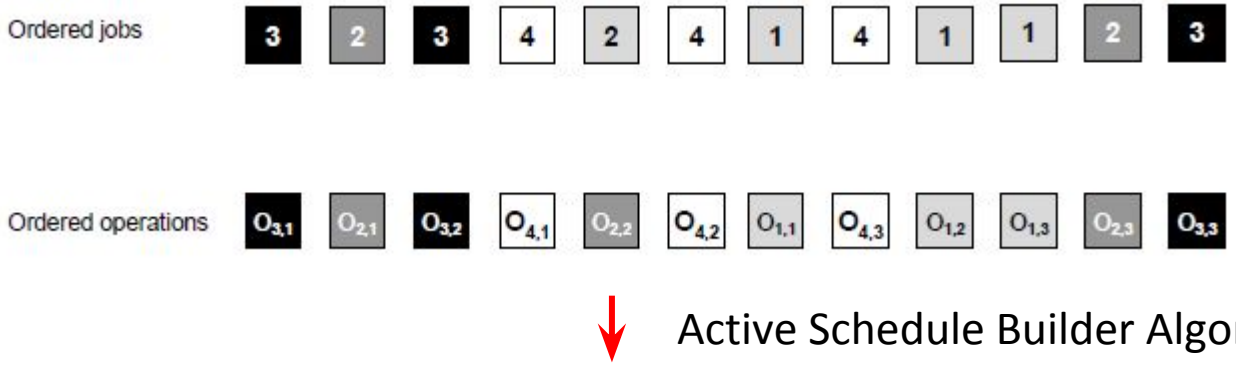
Machine				
a	J1	J4	J2	J3
b	J2	J4	J3	J1
c	J3	J2	J4	J1

But to run Tabu Searches we build an implicit Disjunctive Graph.

Proposed Method

Solution Generator (from BRKGA) :

Active-Schedule build from 1 random job permutation order:



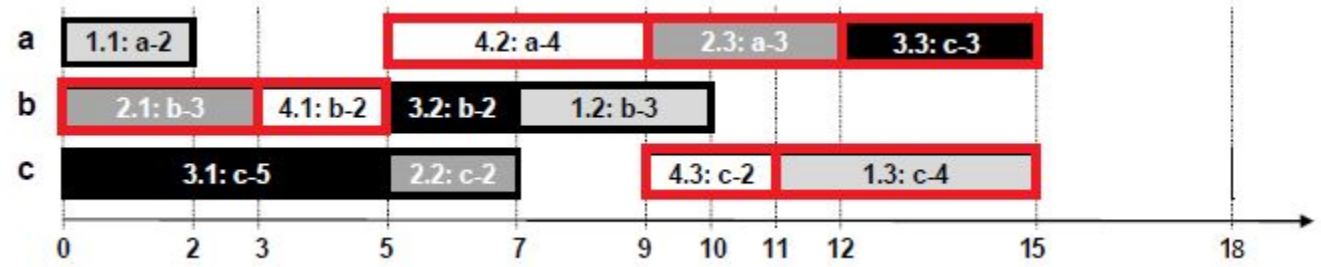
Active Schedule Builder Algorithm

Machine				
a	J1	J4	J2	J3
b	J2	J4	J3	J1
c	J3	J2	J4	J1

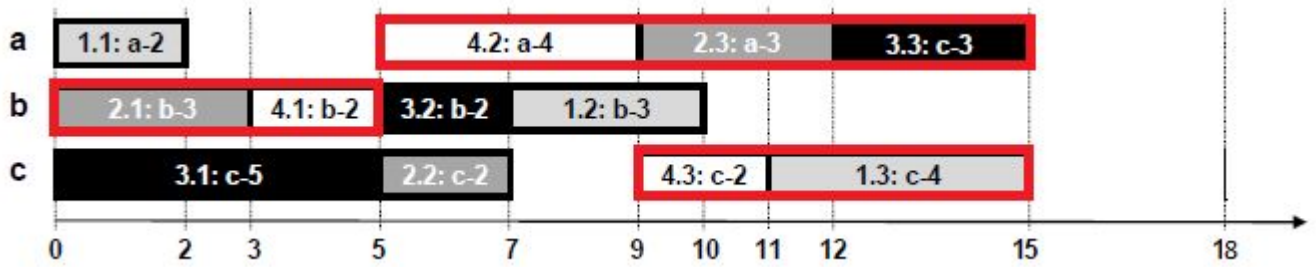
Proposed Method

Tabu Search Concepts: Critical Block

Same machine sequence of operations in a critical path.



Critical Operations.

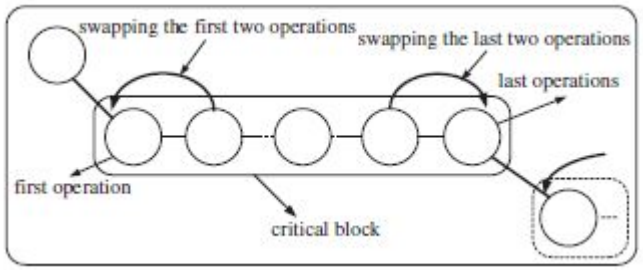


Critical Blocks.

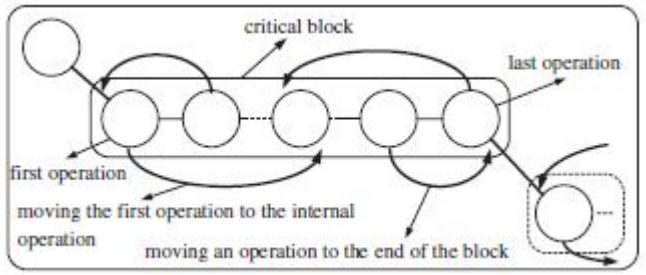
Proposed Method

Tabu Search Concepts: Neighborhoods N5, N6 and N7

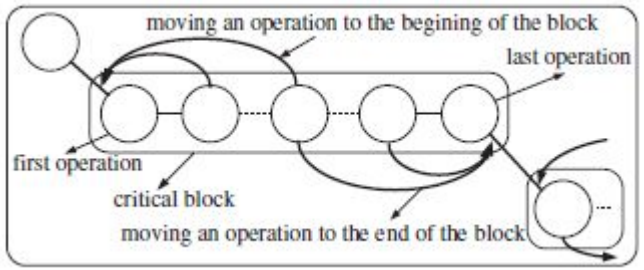
N5



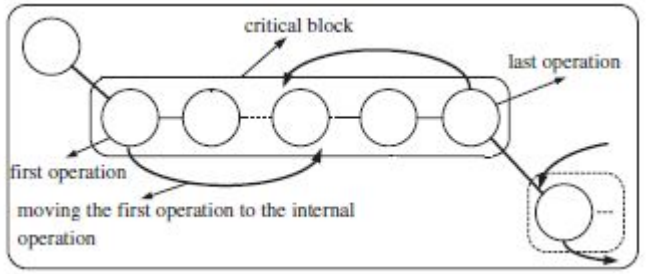
N7



N6



Used Neighborhood



Proposed Method

Tabu Search Concepts: Balas approx. Evaluation

The exact makespan evaluation of a move is time consuming. O(n*m)

We use the Balas[2] approximative strategy that evaluates in O(|Size of the block|).
It fails when there is a critical path not passing by any of the block vertices.

This is the same approach of PR/TS.

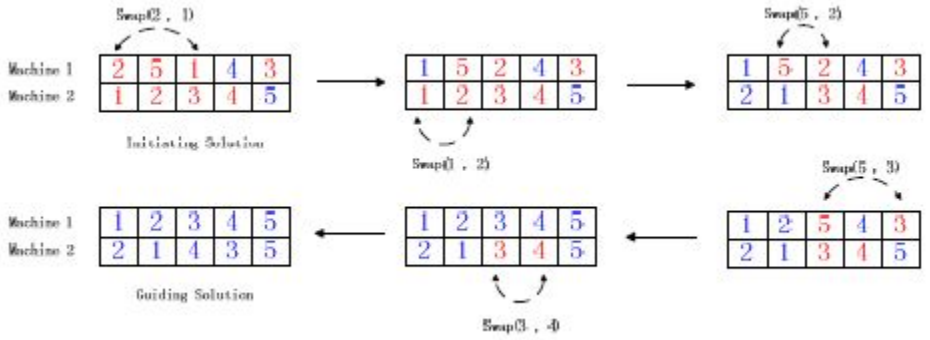
Proposed Method

Path Relink Crossover (from PR/TS):

Given solutions S1 and S2:

Distance between solutions are the number of operations in different machine order.

The path construction algorithm swap the position of a (random chosen) operation that is in wrong position.



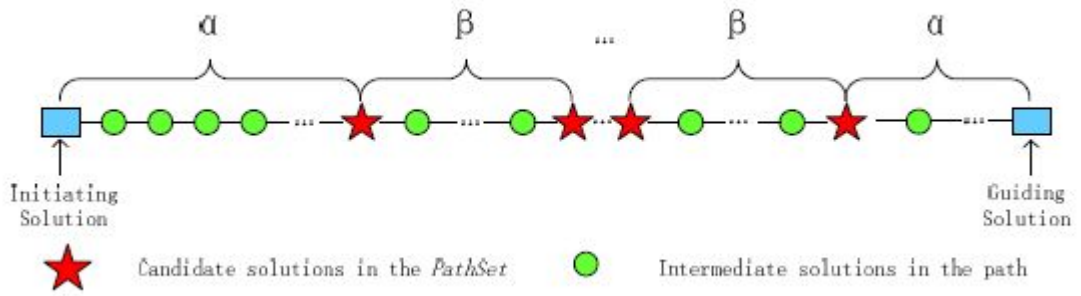
Proposed Method

Path Relink Crossover (from PR/TS):

Given solutions S1 and S2:

Children evaluation is done every “b” steps.

Closest “a” children's are not evaluated.



Proposed Method

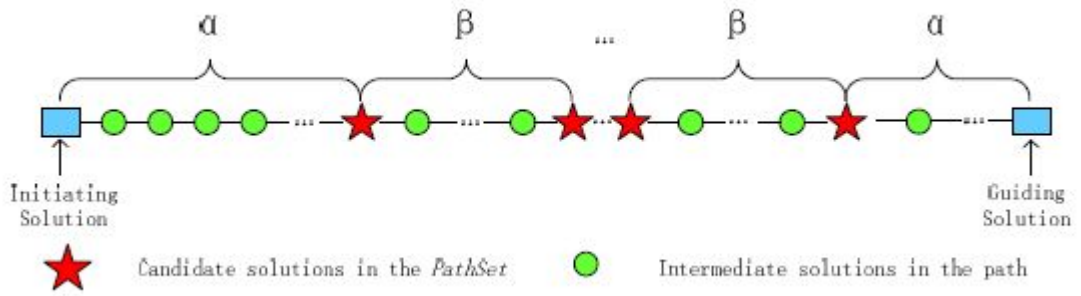
Path Relink Crossover (from PR/TS):

Given solutions S1 and S2:

Candidate children's evaluation is done with a “light” Tabu Search (small iterations limit).

The best evaluated solution is chosen to be children of S1 and S2.

Infeasible solutions are fixed with a simple proposed method.



Proposed Method

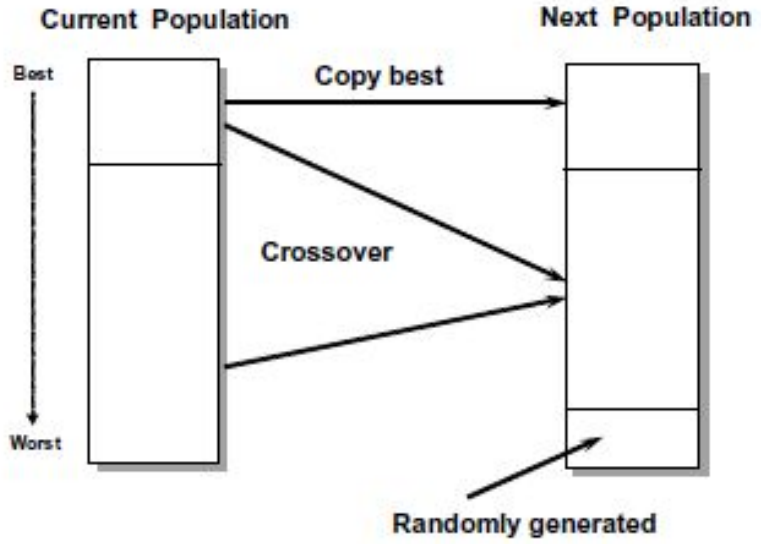
Parents selection (from BRKGA):

We keep selecting random elite solutions and random solutions from population.

For each pair of solutions S1 and S2 we apply the path relink crossover from S1 to S2 and from S2 to S1 to generate two childrens.

Proposed Method

```
P = Initial_Population()
Tabu_Search(P)
While Stop_Criteria_Test()
        Offspring = Generate_Offspring(P)
        Offspring = Tabu_Search(Offspring)
    Mutants = Generate_Random_Solutions()
        Mutants = Tabu_Search(Mutants)
        Population = Elite(Population) + Offspring + Mutants
```



Results

Instances selected are the classic **Taillard 1 - 40**
Ta1-10 : 15x15 | Ta11-20 : 20x15 | Ta21-30 : 20 x 20 | Ta31-40 : 30 x 15

Computational Settings:
Windows 10
Visual Studio 2015
C++
Intel i7-5500U CPU @ 2.40GHz
8 GB RAM

Results

Used Parameters:
Tabu List Size = 2* Number of jobs.
Tabu Search Stop = 100 (high intensive) or 10 (low intensive) non improving iterations.

Population:
Population Size = 100 | Elite Size = 20 | Offspring = 70 | Mutants = 10

Path Relink:
Jump size = dist / 15 | Border Size = dist / 5

Stop Criteria : 20 generations without improve.

Results

Compared to the Best Know Solutions of Literature:

10 Runs of the proposed method.

	ARE	AvgTime [10 Runs]
Ta01-10	0.50%	11.4698
Ta11-20	1.83%	1401.59
Ta21-30	1.34%	27.11458
Ta31-40	1.80%	48.5478
All	1.37%	27.30671

Instance	BestSol	AvgSol	AvgTime	AvgIterations	RE
Ta01	1231	1241.6	12.2986	35.5	0.00%
Ta02	1244	1244.4	10.917	32.9	0.00%
Ta03	1221	1227.4	12.3574	35.2	0.25%
Ta04	1175	1181.6	11.9347	38.4	0.00%
Ta05	1233	1237.4	11.0859	31.9	0.74%
Ta06	1241	1253.6	13.2566	38.9	0.24%
Ta07	1228	1231.2	9.3247	28.6	0.08%
Ta08	1218	1220.2	9.2829	28.3	0.08%
Ta09	1291	1298.4	14.361	44.1	1.33%
Ta10	1269	1269.8	9.8792	31.3	2.26%
Ta11	1399	1408.4	25.9688	55.1	3.10%
Ta12	1383	1385.4	17.5275	38.2	1.17%
Ta13	1370	1384.6	24.4739	54	2.09%
Ta14	1360	1364.8	17.84	32.5	1.12%
Ta15	1357	1373	23.8808	45.7	1.34%
Ta16	1377	1400.3	23.7733	56.1	1.25%
Ta17	1477	1487.7	21.9081	46.8	1.03%
Ta18	1441	1451.7	22.5521	47.1	3.22%
Ta19	1364	1383.2	22.1066	47	2.40%
Ta20	1370	1376.8	20.9155	43.5	1.63%

Results

Compared to the Best Know Solutions of Literature:

10 Runs of the proposed method.

	ARE	AvgTime [10 Runs]
Ta01-10	0.50%	11.4698
Ta11-20	1.83%	1401.59
Ta21-30	1.34%	27.11458
Ta31-40	1.80%	48.5478
All	1.37%	27.30671

Ta21	1673	1687.6	28.2857	46.6	1.89%
Ta22	1613	1622.8	27.1079	46.2	0.81%
Ta23	1575	1588.9	25.5478	42.8	1.16%
Ta24	1654	1656.2	23.6188	40.7	0.61%
Ta25	1625	1643.5	29.9374	46.7	1.88%
Ta26	1685	1693.8	27.4213	47.9	2.43%
Ta27	1696	1711.7	25.4748	44.1	0.95%
Ta28	1634	1641.3	25.55	44.8	1.93%
Ta29	1631	1646	22.8909	40.7	0.37%
Ta30	1606	1617.3	35.3112	55.9	1.39%
Ta31	1771	1777.5	45.1217	61.3	0.40%
Ta32	1836	1854.9	56.8585	70.3	2.91%
Ta33	1834	1852.4	49.4425	68.7	2.40%
Ta34	1892	1919.1	42.2691	56.5	3.44%
Ta35	2007	2009.8	33.1682	41	0.00%
Ta36	1844	1853.4	50.7773	64.2	1.37%
Ta37	1808	1818.3	42.4493	61.3	2.09%
Ta38	1705	1719.4	52.4972	75.7	1.91%
Ta39	1806	1822.9	54.849	65.1	0.61%
Ta40	1721	1743.5	58.0452	82.3	2.81%

Future Work

- Parameter tuning (Tabu Search, Path Relink and Population control)

- Dynamic changes of parameters to intensify or diversify the search.
  - ex. keep track of mean crossover parents distance

- Include more sophisticated procedures from the reference
  - BRKGA Graphical Local Search
  - PR/TS more complex neighborhood

- Use the elite population most common decisions as input of a mathematical programming procedure

References

[1]Bo Peng, Zhipeng Lu, T.C.E. Cheng, 2014,
A Tabu Search/Path Relinking Algorithm to Solve the Job Shop Scheduling Problem.

[2]Balas, E., Vazacopoulos, A., 1998. Guided local search with shifting bottle neck for job shop scheduling. Management Science 44 (2), 262–275.

[3]Gonçalves, J. F., Resende, M. G. C., 2013. An extended akers graphical method with a biased random-key genetic algorithm for job-shop schedul ing. International Transactions in Operational Research.

[4]Nowicki, E., Smutnicki, C., 1996. A fast taboo search algorithm for the job shop problem. Management Science 42 (6), 797–813.

[5]Nowicki, E., Smutnicki, C., 2005. An advanced tabu search algorithm for the job shop problem. Journal of Scheduling 8, 145–159.

[6]Zhang, C. Y., Li, P. G., Guan, Z. L., Rao, Y. Q., 2007. A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem. Computers & Operations Research 34 (11), 3229 – 3242.