2.7 Meta-RaPS Algorithm for the Aerial Refueling Scheduling Problem

Meta-RaPS Algorithm for the Aerial Refueling Scheduling Problem

Sezgin Kaplan; Arif Arin; Ghaith Rabadi
Old Dominion University
Engineering Management and Systems Engineering
skaplan@odu.edu; aarin@odu.edu; grabadi@odu.edu

Abstract. The Aerial Refueling Scheduling Problem (ARSP) can be defined as determining the refueling completion times for each fighter aircraft (job) on multiple tankers (machines). ARSP assumes that jobs have different release times and due dates. The total weighted tardiness is used to evaluate schedule's quality. Therefore, ARSP can be modeled as a parallel machine scheduling with release times and due dates to minimize the total weighted tardiness. Since ARSP is NP-hard, it will be more appropriate to develop approximate or heuristic algorithm to obtain solutions in reasonable computation times. In this paper, Meta-Raps-ATC algorithm is implemented to create high quality solutions. Meta-RaPS (Meta-heuristic for Randomized Priority Search) is a recent and promising metaheuristic that is applied by introducing randomness to a construction heuristic. The Apparent Tardiness Rule (ATC), which is a good rule for scheduling problems with tardiness objective, is used to construct initial solutions which are improved by an exchanging operation. Results are presented for generated instances.

1.0 INTRODUCTION

Resources commonly occur in parallel and many real life problems can be modeled as parallel machine scheduling problems. A parallel machine scheduling problem involves both resource allocation and sequencing. It allocates jobs to each machine and determines the sequence of allocated jobs on each machine. Aerial refueling (AR) is the process of transferring fuel from a tanker aircraft to another receiver aircraft during flight. Aerial refueling scheduling problem (ARSP) aims to determine the starting and completion times of refueling process of each receivers on the tankers. ARSP can be modeled as an identical parallel machine scheduling problem with release times and due dates. It represents a system with m identical machines in parallel and n jobs where job j arrives (becomes available) at ready time r_j and should be complete and leave by the due date d_j. The objective is to find the schedule minimizing total weighted tardiness (TWT) as a performance measure to maintain the quality of service with due dates.

Since ARSP is NP-hard from complexity point of view, it is required to develop effective solution approaches with reasonable computation times. In this study, a fairly new metaheuristic, Meta-RaPS, will be applied to solve the ARSP. Meta-RaPS stands for "Meta-heuristic for Randomized Priority Search", and is one of the randomized search metaheuristics. DePuy et al. [1] expresses the advantages of the Meta-RaPS over other metaheuristics. According to them, run times for Meta-RaPS is not significantly affected by the size of the problem, it is easy to understand and to implement, and can generate a feasible solution at every iteration. It requires a simple dispatching rule to randomize and escape local optima.

Dispatching (or Priority) Rules are the most common heuristics for scheduling problems due to their easy implementation and low computational requirements. The Apparent Tardiness Cost (ATC) heuristic is a good composite dispatching rule for the parallel machine total weighted tardiness problem and is used with MetaRaPS in this paper.

The rest of this paper is organized as follows. In Section 2, the related research is summarized. The Meta-RaPS metaheuristic is explained in Section 3 and the ATC rule in Section 4. A computational study is described in Section 5 by giving an example of the construction phase calculations and a comparison of the TWT values obtained by Meta-RaPS-ATC algorithm with the values obtained by ATC alone. Finally results are concluded in Section 6.

2.0 RELATED WORK

Some researchers addressed scheduling identical parallel machines with ready times to minimize total weighted tardiness problem. Mönch et al. [2] attempted to minimize total weighted tardiness on parallel batch machines with incompatible job families and unequal ready times. They proposed two different decomposition approaches. Dispatching and scheduling rules were used for the batching phase and the sequencing phase of the two approaches. Reichelt et al. [3] were interested in minimizing total weighted tardiness and makespan at the same time. In order to determine a pareto efficient solution for the scheduling of jobs with incompatible families on parallel batch machines problem, they suggested a hybrid multi objective genetic algorithm. Pfund et al. [4] addressed scheduling jobs with ready times on identical parallel machines with sequence dependent setups by minimizing the total weighted tardiness. Their approach was an extension of the Apparent Tardiness Cost with Setups (ATCS) approach by Lee and Pinedo [5] to allow non-ready jobs to be scheduled. Gharehgozli et al. [6] presented a new mixed-integer goal programming (MIGP) model for a parallel machine scheduling problem with sequence-dependent setup times and release dates. Fuzzy processing times and two fuzzy objectives were considered in the model to minimize the total weighted flow time and the total weighted tardiness simultaneously.

There are also a few Meta-RaPS applications on scheduling problems. Hepdogan et al. [7] investigated Meta-RaPS approach to the single machine early/tardy scheduling problem with common due date and sequence-dependent setup times. The objective of their problem was to minimize the total amount of earliness and tardiness of jobs that are assigned to a single machine. Rabadi et al. [8] introduced Meta-RaPS approach to the non-preemptive unrelated parallel machine scheduling problem with the objective of minimizing the makespan. In their problem, machine-dependent and job sequence-dependent setup times were considered when all jobs are available at time zero, and all times are deterministic.

3.0 THE META-RAPS ALGORITHM

Moraga et al. [9] defines Meta-RaPS as "generic, high level search procedures that introduce randomness to a construction heuristic as a device to avoid getting trapped at a local optimal solution". Meta-RaPS combines the mechanisms of priority rules, randomness, and sampling.

A Meta-RaPS algorithm uses four parameters: the number of iterations (*I*), the priority percentage (p%), the restriction percentage (*r*%), and the improvement percentage (*i*%). Meta-RaPS does not select the component or activity with the best priority value every time, nor the incremental cost. However, the algorithm may accept one with a good priority value, not necessarily the best, based on a randomized approach. The parameter *p*% is employed to decide the percentage of time, the component, or activity with the best priority value will be added to the current partial solution, and 100%-*p*% of time the component or activity with the good priority value is randomly selected from a candidate list (CL) containing "good" components or activities. The CL of components or activities with good priority values is created by including ones whose priority values are within *r*% of the best priority value.

Meta-RaPS is a two-phase metaheuristic: a constructive phase to create feasible solutions and an improvement phase to improve them. In the constructive phase, a solution is built by repeatedly adding feasible components or activities to the current solution in order based on their priority rules until the stopping criterion is satisfied. Generally, solutions obtained by implementing only constructive algorithms can reach mostly local optima. To avoid local optima, Meta-RaPS employs randomness in the constructive phase so that solutions other than the best solution can be selected.

The improvement phase is performed if the feasible solutions generated in the construction phase are within *i*% of the best unimproved solution value from the preceding iterations [9].

4.0 ATC RULE

The Apparent Tardiness Cost (ATC) will be applied to the parallel machine total weighted tardiness problem as a composite dispatching rule. Pinedo [10] defines a composite dispatching rule as "a ranking expression that combines a number of elementary dispatching rules". An elementary rule is a function of constant or time dependent properties of the jobs and/or the machines, i.e. processing times, due dates for jobs; speed, number of jobs waiting for processing for machines, etc. The ATC combines the elementary Weighted Shortest Processing Time first (WSPT) dispatching rule and the Minimum Slack first (MS) rule. According to the WSPT rule the jobs are ordered in decreasing order of w_j/p_j; and the MS rule selects at time t, when a machine is freed, among the remaining jobs the job with the minimum slack where the slack can be defined as $max\ (d_j - p_j - t,\ 0)$. Every time the machine becomes free, the ATC calculates a ranking index for each remaining job. The job with the highest ranking index defined in equation 1 is then selected to be processed next:

$$I_j(t) = \frac{w_j}{p_j} \exp\left(-\frac{\max(d_j - p_j - t, 0)}{K\bar{p}} \right) \qquad (1)$$

where $\bar{p}$ is the average processing time of the remaining jobs, and K is the scaling parameter, called look-ahead parameter. If K is very large the ATC rule behaves similar to the WSPT rule, and if K is very small the rule behaves similar to the MS rule. The WSPT rule is optimal when all jobs are tardy, while the MS rule is optimal when all due dates are sufficiently loose and spread out.

The effectiveness of the ATC heuristic depends on the value of the look-ahead parameter K. Previous studies have usually recommended a fixed value of K between 0.5 and 2.0 [11].

5.0 COMPUTATIONAL STUDY

5.1 Parameter Setting

The accepted values of the parameters to be employed in metaheuristics have a significant impact on both the solution process and solution quality. Particularly, in terms of the interactions, Design of Experiments (DOE) methods are promising approaches and can be employed to tune the parameters more effectively. In this study, we applied 3-level (3^k) full factorial design to tune the parameters of Meta-RaPS. After completing regression analysis with R^2 = 0.95, the values found for the parameters are presented in Table 1.

Table 1. Meta-RaPS Parameter Setting

Parameter	Value
Number of iterations (I)	10000
Priority percentage (p%)	25%
Restriction percentage (r%)	60%
Improvement percentage (i%)	70%

5.2 Meta-RaPS-ATC Algorithm

To present the effectiveness of Meta-RaPS-ATC algorithm, problems were solved both by using ATC rule and Meta-RaPS-ATC approach with the tuned parameters. In the ATC, the jobs are selected by calculating their ATC index, and the one with the highest index is always selected. However, in Meta-RaPS-ATC algorithm, the ATC index for each job is calculated, and the selection is made based on Meta-RaPS principles. If the random number (RN) is smaller or equal to the priority percentage, the job with the highest ATC index is selected. If not, a lower limit is calculated by multiplying the highest index by the restriction percentage. Jobs whose ATC indices are higher than the lower limit are added to the CL, and the next job is selected from this CL randomly. After all jobs are assigned to machines, the construction phase of Meta-RaPS is completed. For both algorithms, the ATC indices are updated after the selection of each job.

In Meta-RaPS, only the constructed solutions with promising, or good enough, values are improved. To determine this level, summation of the lowest (best) solution with the multiplication of the difference between the highest (worst) solution value and the best solution value obtained until current iteration by the improvement percentage is used. If the current solution value is higher than this level, the improvement phase is performed by swapping two arbitrarily selected jobs in the constructed schedule and comparing with the best and the worst solution values in memory. After swapping operation, jobs are scheduled by taking into account the release time of the swapped job and the completion time of the predecessor job.

5.3 Results for ARSP

To simulate ARSP, we randomly generated 10 instances with release times, processing times, weights and due dates for m = 3 machines and n = 12 jobs so that an optimal solution can be obtained in a reasonable time.

One of these instances whose data is given in Table 2. is used as an example to explain Meta-RaPS.

Table 2. Data for Example Problem

Job	Release Time	Processing Time	Weight	Due Date
1	19	30	4	79
2	1	28	7	55
3	24	23	8	70
4	1	28	8	55
5	3	19	6	41
6	10	45	5	100
7	2	45	4	92
8	5	43	3	91
9	11	28	1	67
10	0	23	8	46
11	13	29	1	71
12	15	32	2	79

The construction phase of Meta-RaPS algorithm only for one iteration is shown in Table 3. In every step of this phase, jobs are assigned to the machines with the earliest availability. The total weighted tardiness of this solution is 250.

A Mixed Integer Linear Programming (MILP) model was developed to find optimal solutions. Optimization Programming Language (OPL) Studio 6.3 was used to implement this model and CPLEX 12.1 to solve it.

Table 3. Construction Phase of Meta-RaPS ATC Algorithm

Step	Machine 1	Machine 2	Machine 3	Max. ATC	Job	Lower Limit	CL	RN	RN > p	Assignment	Job	Machine
1	0	0	0	0.187	10	0.112	4, 7	0,21	NO	max	10	1
2	23	0	0	0,177	5	0,106	2,4	0,85	YES	CL	2	2
3	23	28	0	0,178	5	0,107	4	0,77	YES	CL	4	3
4	23	28	28	0,316	5	0,189	empty	-	-	max	5	1
5	42	28	28	0,219	3	0,132	empty	-	-	max	3	2
6	42	51	28	0,082	1	0,049	6,7	0,51	YES	CL	7	3
7	42	51	73	0,113	1	0,068	6,7	0,16	NO	max	1	1
8	72	51	73	0,101	6	0,061	8	0,23	NO	max	6	2
9	72	96	73	0,021	8	0,012	12	0,69	YES	CL	12	1
10	104	96	73	0,073	8	0,044	empty	-	-	max	8	3
11	104	96	116	0,036	9	0,021	11	0,22	NO	max	9	2
12	104	124	116	-	-	-	-	-	-	-	11	1

Table 4. Comparison of ATC and Meta-RaPS ATC Algorithm

Instance	ATC	Meta-RaPS	Optimal	ATC Deviation from Optimal	Meta-RaPS-ATC Deviation from Optimal
1	707.5	495.3	471.5	0.50	0.05
2	248.0	218.6	216.0	0.15	0.01
3	790.0	770.0	680.0	0.16	0.13
4	270.9	265.8	238.0	0.14	0.12
5	478.1	364.0	363.0	0.32	0.00
6	484.0	394.5	372.0	0.30	0.06
7	276.0	278.6	276.0	0.00	0.01
8	211.5	116.6	100.5	1.10	0.16
9	315.7	296.0	287.0	0.10	0.03
10	526.2	364.6	348.0	0.51	0.05
			Average	**0.32**	**0.06**

The results for 10 instances are summarized in Table 4. While the average deviation of the ATC results from the optimal solutions was 0.32, and the average deviation of the Meta-RaPS-ATC algorithm solutions was 0.06. Based on the findings for the ARSP instances, using Meta-RaPS-ATC approach gives better results than using ATC rule.

6.0 CONCLUSIONS

ARSP is a real world problem that requires high quality solutions in an acceptable time frame. As the dimensions of the problem get larger, the solution process of mathematical modeling loses its effectiveness. Using only composite dispatching rules, such as the ATC rule, may not give the best solutions for most applications. However, metaheuristics can offer high quality solutions, and Meta-RaPS seems to be a promising metaheuristic with its simplicity and effectiveness to find high quality solutions for ARSP, and for scheduling problems in general.

More computation and analysis are needed for better performance comparisons in instances with large number of jobs. In future research, more constraints such as machine compatibility, sequence dependent setup times and deadlines may be included in the model.

7.0 REFERENCES

1. DePuy G. W., Whitehouse G. E. and Moraga R. J., Meta-RaPS: A Simple And Efficient Approach For Solving Combinatorial Problems, 29th International Conference on Computers and Industrial Engineering, November 1-3, Montreal, Canada, 644-649, 2001.

2. Mönch, L., Balasubramanian, H., Fowler, J. W. and Pfund, M.E., Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times, Computers & Operations Research, 32, 2731–2750, 2005.

3. Reichelt, D. , Mönch, L., Gottlieb, J. and Raidl, G.R., Multiobjective Scheduling of Jobs with Incompatible Families on Parallel Batch Machines, EvoCOP 2006, LNCS 3906, Berlin Springer-Verlag Heidelberg, 209–221, 2006.

4. Pfund, M., Fowler, J. W., Gadkari, A. and Chen, Y., Scheduling jobs on parallel machines with setup times and ready times, Computers and Industrial Engineering, 54, 764–782, 2008.

5. Lee, Y.H. and Pinedo, M., Theory and methodology: Scheduling jobs on parallel machines with sequence-dependent setup times, European Journal of Operational Research 100, 464-474, 1997.

6. Gharehgozli, A.H., Tavakkoli, R. and Zaerpour, N., A fuzzy-mixed-integer goal programming model for a parallel-machine scheduling problem with sequence-dependent

setup times and release dates, Robotics and Computer-Integrated Manufacturing, 25, 853–859, 2009.

7. Hepdogan S., Moraga R., DePuy G.W., and Whitehouse G.E, A Meta-RaPS for the early/tardy single machine scheduling problem, International Journal of Production Research, Vol. 47, No. 7, 1717–1732, 2009.

8. Rabadi G., Moraga R.J. and Al-Salem A., Heuristics for the unrelated parallel machine scheduling problem with setup times, Journal of Intelligent Manufacturing, 17, 85–97, 2006.

9. Moraga R. J., DePuy G. W. and Whitehouse G. E., Metaheuristics: A Solution Methodology for Optimization Problems, Handbook of Industrial and Systems Engineering, CRC Press, FL, 2006.

10. Pinedo M. L., Scheduling Theory, Algorithms, and Systems, Third Edition, Springer, NY, 2008.

11. Holsenback J. E., Russell R. M., Markland R. E. and Philipoom P. R., An improved heuristic for the single-machine, weighted-tardiness problem, Omega 27, 485–495, 1999.

Meta-RaPS Algorithm for the Aerial Refueling Scheduling Problem

Sezgin KAPLAN, Ph.D. Student
Old Dominion University

October 13, 2010

MODSIM WORLD Conference & Expo

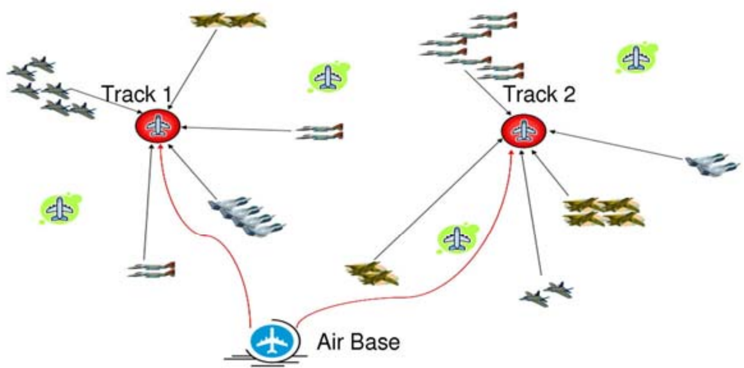
Outline

- Aerial Refueling Scheduling Problem (ARSP)
- ARSP Solution Method
- Related Works
- Meta-RaPS Algorithm
- Apparent Tardiness Cost (ATC) Rule
- Computational Study
- Conclusion
- References

ARSP

Aerial Refueling: The process of transferring fuel from one aircraft (a tanker) to another (a fighter as a receiver) during flight.

ARSP



- ARSP: Determining the refueling completion times for each fighter aircraft (job) on one of multiple tankers (machines).

ARSP

- **Model :** An identical parallel machine scheduling problem with release times and due dates.
- A system with *m* identical machines in parallel and *n* jobs where job *j* becomes available at ready time r_j and should be complete and leave by the due date d_j.
- **Objective :** To minimize total weighted tardiness (TWT) as a performance measure.

Solution Method

- MetaRaPS- ATC algortihm will be applied to solve the ARSP.
- "Meta-heuristic for Randomized Priority Search" (Meta-RaPS) is;
  - One of the randomized search metaheuristics.
  - It requires a simple dispatching rule to randomize and escape local optima [1].
- The Apparent Tardiness Cost (ATC) heuristic is used as a dispatching rule with MetaRaPS.

MODSIM WORLD Conference & Expo

Related Works

- **Hepdogan et al. [7]**

  Problem: The single machine early/tardy scheduling problem with common due date and sequence-dependent setup times.

  Objective : To minimize the total amount of earliness and tardiness of jobs.

- **Rabadi et al. [8]**

  Problem: The non-preemptive unrelated parallel machine scheduling problem with machine-dependent and job sequence-dependent setup times when all jobs are available at time zero, and all times are deterministic.

  Objective : To minimize the makespan.

MODSIM WORLD Conference & Expo

Meta-RaPS

- Generic, high level search procedure that introduce randomness to a construction heuristic as a device to avoid getting trapped at a local optimal solution [9].
- Combination of the mechanisms of priority rules, randomness, and sampling.
- Meta-RaPS advantages over other metaheuristics;
  - Run time is not significantly affected by the size of the problem,
  - Easy to understand and to implement,
  - Able to generate a feasible solution at every iteration [1].

Meta-RaPS

Two-phase in each iteration:

1. **Constructive phase** to create feasible solutions.
   – A solution is built by repeatedly adding feasible activities to the current solution in order based on their priority rules.
   – To avoid local optima, randomness is employed so that solutions other than the best solution can be selected.
2. **Improvement phase** to improve the feasible solution.
   – Only the constructed solutions with promising, or good enough, values are improved.
   – A neighborhood search is performed.

Meta-RaPS

Meta-RaPS algorithm uses four parameters:

1. **Priority percentage** (p%): Decision to schedule the best or randomly one from candidate list.
2. **Restriction percentage** (*r*%) : Determining the CL by calculating a lower limit for good priority values. The lower limit is *r*% of the best priority value.
3. **Improvement percentage** (*i*%) : Decision to improve or not improve the constructed solution.
4. **Number of iterations** (*I*) : Stopping criterion

MODSIM WORLD Conference & Expo

Meta-RaPS

The pseudocode for one iteration of the basic Meta-RaPS procedure is given as follows:

Construction Phase:

```
1.  Do until feasible solution generated
2.      Find priority value for each feasible activity
3.      Find best ATC priority value
4.      P = RND(1,100)
5.      If P ≤ %priority Then
6.          Add activity with best priority value to solution
7.      Else
8.          Form "available" list of all feasible activities whose priority values
             are within %restriction of best priority value
9.          Randomly choose activity from available list and add to solution
10.     End If
11. End Until
```

Improvement Phase:

```
12. If an iteration's solution value is within %improvement of the best solution
    value found so far, Then
13.     A neighborhood search is performed
14. End If
15. Calculate and Print solution
```

MODSIM WORLD Conference & Expo

ATC Rule

- A good composite dispatching rule for the total weighted tardiness problem [4].
- A dynamic algorithm that after each job completion, the remaining jobs are analyzed, priorities derived according to improved formula, and the highest priority job selected.
- The priority index :

$$\pi_j(t) = \frac{w_j}{p_j}\exp\left[-\frac{\max\left[d_j - p_j - t, 0\right]}{k\overline{p}}\right]$$

  - Urgent job :
    If $C_j \geq d_j$, then $\pi_j = w_j/p_j$

w_j : weight of the remaining job j
p_j : processing time of the job j
d_j : due date of the job j
t : Decision time point
$\overline{p}$: The average processing time of the remaining jobs,
k : 'look-ahead' parameter.
$S_j^+(t)$, the slack factor : $max\ [d_j - p_j - t, 0]$.

Parameter Setting

- The accepted values of the parameters to be employed in metaheuristics have a significant impact on both the solution process and solution quality.
- Design of Experiments (DOE) methods are used to tune the parameters.

Table 3. Meta-RaPS Parameters

Parameter	Value
Number of iterations (I)	10000
Priority percentage (p%)	25%
Restriction percentage (r%)	60%
Improvement percentage (i%)	70%

An Example

Table 1. Data for Example Problem (m = 3 machines and n = 12 jobs)

Job	Release Time	Processing Time	Weight	Due Date
1	19	30	4	79
2	1	28	7	55
3	24	23	8	70
4	1	28	8	55
5	3	19	6	41
6	10	45	5	100
7	2	45	4	92
8	5	43	3	91
9	11	28	1	67
10	0	23	8	46
11	13	29	1	71
12	15	32	2	79

An Example

Table 2. Construction Phase of Meta-RaPS ATC Algorithm

Step	Machine 1	Machine 2	Machine 3	Max. ATC	Job	Lower Limit r= 0.6	CL	RN	RN > p p= 0.25	Assignment	Job	Machine
1	0	0	0	0.187	10	0.112	4,7,10	0.21	NO	max	10	1
2	23	0	0	0,177	5	0,106	2,4,5	0.85	YES	CL	2	2
3	23	28	0	0,178	5	0,107	4,5	0.77	YES	CL	4	3
4	23	28	28	0,316	5	0,189	5	-	-	max	5	1
5	42	28	28	0,219	3	0,132	3	-	-	max	3	2
6	42	51	28	0,082	1	0,049	1,6,7	0.51	YES	CL	7	3
7	42	51	73	0,113	1	0,068	1,6,7	0.16	NO	max	1	1
8	72	51	73	0,101	6	0,061	6,8	0.23	NO	max	6	2
9	72	96	73	0,021	8	0,012	8,12	0.69	YES	CL	12	1
10	104	96	73	0,073	8	0,044	8	-	-	max	8	3
11	104	96	116	0,036	9	0,021	9,11	0.22	NO	max	9	2
12	104	124	116	-	-	-	-	-	-	-	11	1

Machine 1: 10-5-1-12-11 **Machine 2**: 2-3-6-9 **Machine 3** : 4-7-8 **TWT** : 250

Assume Best : 200, Worst : 400, *%i* : 0.7
Decision level: 200 + (400-200)*0.7 = 340 > TWT Improve

Improvement : Swapping two arbitrarily selected jobs in the constructed schedule by taking into account the release times.

Computational Study

- To present the effectiveness of Meta-RaPS algorithm, problems were solved both by using ATC rule and Meta-RaPS algorithm with the tuned parameters.
- To simulate ARSP, 10 instances with release times, processing times, weights and due dates were randomly generated.
- A Mixed Integer Linear Programming (MILP) model was developed to find optimal solutions. Optimization Programming Language (OPL) Studio 6.3 was used to implement this model and CPLEX 12.1 to solve it.

Computational Study

Table 4. Comparison of ATC and Meta-RaPS ATC Algorithm

Instance	ATC	Meta-RaPS	Optimal	ATC Deviation from Optimal	Meta-RaPS-ATC Deviation from Optimal
1	707.5	495.3	471.5	0.50	0.05
2	248.0	218.6	216.0	0.15	0.01
3	790.0	770.0	680.0	0.16	0.13
4	270.9	265.8	238.0	0.14	0.12
5	478.1	363.0	363.0	0.32	0.00
6	484.0	394.5	372.0	0.30	0.06
7	276.0	278.6	276.0	0.00	0.01
8	211.5	116.6	100.5	1.10	0.16
9	315.7	296.0	287.0	0.10	0.03
10	526.2	364.6	348.0	0.51	0.05
			Average	**0.32**	**0.06**

Conclusion

- Using only composite dispatching rules, such as the ATC rule, may not give the best solutions for most applications.
- Meta-RaPS seems to be a promising metaheuristic with its simplicity and effectiveness to find high quality solutions for ARSP.
- More computation and analysis are needed for better performance comparisons in instances with large number of jobs.
- In future research, constraints such as machine compatibility, sequence dependent setup times and deadlines may be included in the model.

References

1. DePuy G. W., Whitehouse G. E. and Moraga R. J., Meta-RaPS: A Simple And Efficient Approach For Solving Combinatorial Problems, 29th International Conference on Computers and Industrial Engineering, November 1-3, Montreal, Canada, 644-649, 2001.
2. Mönch, L., Balasubramanian, H., Fowler, J. W. and Pfund, M.E., Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times, Computers & Operations Research, 32, 2731–2750, 2005.
3. Reichelt, D. , Mönch, L., Gottlieb, J. and Raidl, G.R., Multiobjective Scheduling of Jobs with Incompatible Families on Parallel Batch Machines, EvoCOP 2006, LNCS 3906, Berlin Springer-Verlag Heidelberg, 209–221, 2006.
4. Pfund, M., Fowler, J. W., Gadkari, A. and Chen, Y., Scheduling jobs on parallel machines with setup times and ready times, Computers and Industrial Engineering, 54, 764–782, 2008.
5. Lee, Y.H. and Pinedo, M., Theory and methodology: Scheduling jobs on parallel machines with sequence-dependent setup times, European Journal of Operational Research 100, 464-474, 1997.

References

6. Gharehgozli, A.H., Tavakkoli, R. and Zaerpour, N., A fuzzy-mixed-integer goal programming model for a parallel-machine scheduling problem with sequence-dependent setup times and release dates, Robotics and Computer-Integrated Manufacturing, 25, 853–859, 2009.
7. Hepdogan S., Moraga R., DePuy G.W., and Whitehouse G.E, A Meta-RaPS for the early/tardy single machine scheduling problem, International Journal of Production Research, Vol. 47, No. 7, 1717–1732, 2009.
8. Rabadi G., Moraga R.J. and Al-Salem A., Heuristics for the unrelated parallel machine scheduling problem with setup times, Journal of Intelligent Manufacturing, 17, 85–97, 2006.
9. Moraga R. J., DePuy G. W. and Whitehouse G. E., Metaheuristics: A Solution Methodology for Optimization Problems, Handbook of Industrial and Systems Engineering, CRC Press, FL, 2006.
10. Pinedo M. L., Scheduling Theory, Algorithms, and Systems, Third Edition, Springer, NY, 2008.
11. Holsenback J. E., Russell R. M., Markland R. E. and Philipoom P. R., An improved heuristic for the single-machine, weighted-tardiness problem, Omega 27, 485–495, 1999.

MODSIM WORLD Conference & Expo

Questions/Comments