A Matheuristic for the Generalized Order Acceptance and Scheduling Problem

Ceyda Oğuz

(*joint work with* İstenç Tarhan)

Koç University

Scheduling Seminar Series
February 7, 2024

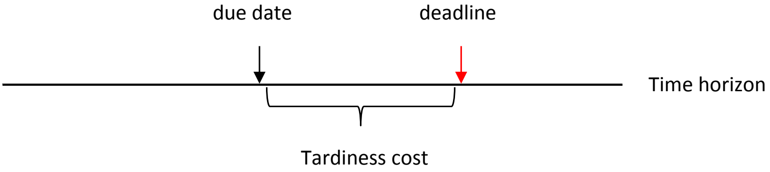
Agenda

- Problem definition
- Literature review
- MILP model
- Matheuristic algorithm
- Computational study
- Conclusion

Problem definition

Order acceptance and scheduling problem

- Make-to-order systems
- Limitations: Capacity and time
- Customer preferences: Acceptable date and deadline



- Integration of two decision:
  - Order Acceptance : Which orders to be accepted?
  - Scheduling Problem : How to schedule them?

Literature review

- Wester et al. (1992), Slotnick and Morton (1996), Rom and Slotnick (2009), Lewis and Slotnick (2002): No deadline
- Charnsirisakskul et al. (2004): Due date and deadline distinction
- Oguz et al. (2010): Extension by sequence dependent setup time and release time (GOAS)
  - MILP model formulation
  - Simulated annealing based heuristic and two constructive heuristics
- Studies following Oguz et al. (2010):
  - Cesaret et al. (2012): Tabu search
  - Park et al. (2013): Genetic algorithm
  - Lin and Ying (2013): Artificial bee colony
  - Chaurasia and Singh (2016):Two hybrid metaheuristics
  - Nguyen (2016): Hyper-heuristic using genetic algorithm
  - Silva et al. (2018): Branch & price and iterated local search algorithm
  - He et al. (2019): Memetic algorithm
  - de Weerdt et al. (2021): Heuristic algorithms based on dynamic programming
- Exact algorithms for similar problems: Slotnick and Morton (2007), Mestry et al. (2009), Nobibon and Leus (2011), Geramipour et al. (2017), Cordone et al. (2017) and de Weerdt et al. (2021)

Indices, sets and parameters

n : the number of orders

$O = \{1, ..., n\}$: the set of orders

$O' = O \cup \{0, n+1\}$: the set of orders including dummies

$i, j \in O'$

$s_{i,j}$: Setup time between order i and order j

p_i : Processing time of order i

e_i : Revenue of order i

r_i : Release time of order i

d_i : Due date of order i

$\bar{d}_i$: Deadline of order i

w_i : Unit tardiness cost of order i

Decision variables

I_i: 1 if order i is accepted, 0 otherwise, $\forall i \in O'$

$y_{i,j}$: 1 if order j is processed immediately after order i, 0 otherwise, $\forall i \in O' \setminus \{n+1\}, j \in O' \setminus \{0\}$

$g_{i,j}$: Completion time of order j if order i immediately precedes it, 0 otherwise, $\forall i \in O' \setminus \{n+1\}, j \in O' \setminus \{0\}$

$T_{i,j}$: Tardiness of order j if order i immediately precedes it, 0 otherwise, $\forall i \in O' \setminus \{n+1\}, j \in O' \setminus \{0\}$

$$\text{Maximize} \quad \sum_{i \in O} e_i I_i - \sum_{i,j \in O'} T_{i,j} w_j \tag{1}$$

s.t.

$$\sum_{j=1, j \neq i}^{n+1} y_{i,j} = I_i \qquad \forall i \in O' \setminus \{n+1\} \tag{2}$$

$$\sum_{j=0, j \neq i}^{n} y_{j,i} = I_i \qquad \forall i \in O' \setminus \{0\} \tag{3}$$

$$\sum_{j=1}^{n+1} g_{i,j} - \sum_{j=0}^{n} g_{j,i} \geq \sum_{j=1}^{n+1} (s_{i,j} + p_j) y_{i,j} + \sum_{j=1, \bar{d}_i < r_j}^{n+1} (r_j - \bar{d}_i) y_{i,j} \qquad \forall i \in O \tag{4}$$

$$g_{i,j} \geq g_{i,j}^{lb} y_{i,j} \qquad \forall i \in O' \setminus \{n+1\}, j \in O' \setminus \{0\} \tag{5}$$

$$g_{i,j} \leq g_{i,j}^{ub} y_{i,j} \qquad \forall i \in O' \setminus \{n+1\}, j \in O' \setminus \{0\} \tag{6}$$

$$T_{i,j} \geq g_{i,j} - d_j y_{i,j} \qquad \forall i \in O' \setminus \{n+1\}, j \in O' \setminus \{0\} \tag{7}$$

$$T_{i,j}, g_{i,j} \in R^+, \; I_i, y_{i,j} \in \{0,1\} \qquad \forall i \in O', j \in O' \tag{8}$$

where

$$g_{i,j}^{lb} = \max(r_i + \min_{k: k \leq n, k \neq i} s_{k,i} + p_i + s_{i,j} + p_j, r_j + s_{i,j} + p_j) \quad \forall i \in O' \setminus \{n+1\}, j \in O' \setminus \{0\}$$

$$g_{i,j}^{ub} = \min(\bar{d}_j, \max(\bar{d}_i, r_j) + s_{i,j} + p_j) \quad \forall i \in O' \setminus \{n+1\}, j \in O' \setminus \{0\}$$

- The GOAS problem is NP-hard
- Performance of the commercial solvers significantly deteriorates as the number of orders increases
- Motivation: Decompose the original problem into subproblems having manageable number of orders
- Proposed algorithm is a matheuristic comprising of two different MILP models (ASSIGN and SCHEDULE) running in an iterative manner
    - ASSIGN: Assign orders to particular segments of the planning horizon by excluding sequence dependent setup times
    - SCHEDULE: Solve the GOAS problem under the restrictions provided by ASSIGN

Matheuristic algorithm GOAS_I2PM

General scheme

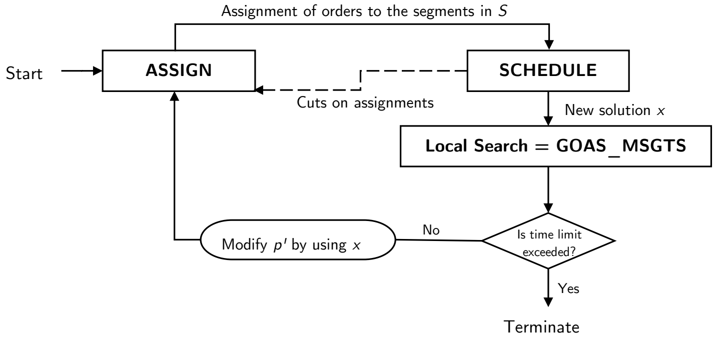


Figure 1: Flow chart of the matheuristic algorithm

- ASSIGN is based on a time index set $T' = \{t_0, t_1, \ldots, t_{|T'|}\}$
- Intervals between adjacent two indices are denoted by index k such that kth interval starts from $t_{k-1} + 1$ and ends at t_k where $k \in K = \{1, .., |T'|\}$ and $t_0 = 0$ w.l.o.g.
- Time indices are generated such that if an order is started in kth interval, it must be completed in $k + 1$st interval at the latest for any $k \in K$

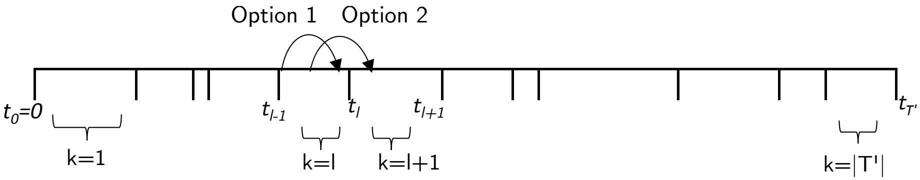


Figure 2: Planning horizon T with time indices

- Original planning horizon T is decomposed into a set of segments S
- The end points of the segments, $\varsigma_1, \varsigma_2, .., \varsigma_{|S|}$, in set S is chosen from the set of time indices T'
- Time segments are denoted by index s such that sth segment spans the time period from ς_{s-1} to ς_s where $s \in S = \{1, 2, .., |S|\}$.

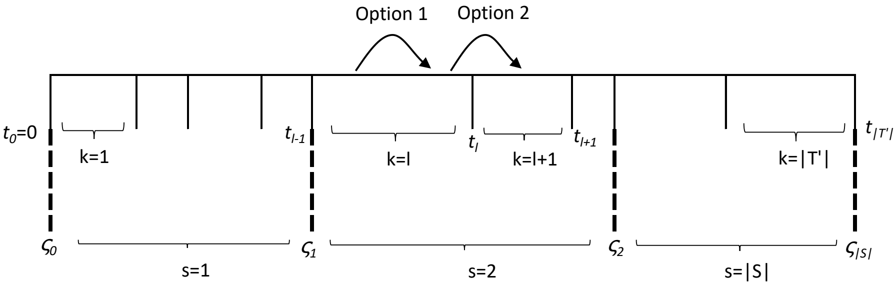


Figure 3: Decomposing the planning horizon T into time segments

Decision variables

$u_{i,k}$: 1 if order i has started and completed in interval k, 0 otherwise, $\forall i \in O, k \in K$

$u'_{i,k}$: 1 if order i has started in interval $k-1$ and completed in interval k, 0 otherwise, $\forall i \in O, k \in K \setminus \{0\}$

$a_{i,s}$: 1 if the completion time of order i lies in time segment s, 0 otherwise, $\forall i \in O, s \in S$

$q_{i,k}$: Amount of order i processed in interval k if $u'_{i,k} = 1$, 0 otherwise, $\forall i \in O, k \in K \setminus \{0\}$

T_i: Tardiness of order i, $\quad \forall i \in O$

$$\text{Maximize} \quad \sum_{i \in O, k \in K} (u_{i,k} + u'_{i,k}) e_i - \sum_{i \in O} T_i w_i \tag{9}$$

s.t.

$$\sum_{i \in O, k \in K} ((u_{i,k} + u'_{i,k+1}) p'_i + q_{i,k} - q_{i,k+1}) \leq t_k - t_{k-1} \qquad \forall k \in K \tag{10}$$

$$\sum_{i \in O} u'_{i,k} \leq 1 \qquad \forall k \in K \tag{11}$$

$$\sum_{i \in O} q_{i,k} \leq u'_{i,k} p'_i \qquad \forall k \in K \tag{12}$$

$$\sum_{k \in K : \varsigma_{s-1} < t_k <= \varsigma_s} (u_{i,k} + u'_{i,k}) = a_{i,s} \qquad \forall i \in O, s \in S \setminus \{0\} \tag{13}$$

$$\sum_{s \in S} a_{i,s} = 1 \qquad \forall i \in O \tag{14}$$

$$T_i \geq \sum_{k \in K} (u'_{i,k}(t_{k-1} - d_i) + q_{i,k}$$

$$+ u_{i,k} max\{max\{t_{k-1}, r_i\} + p'_i - d_i, 0\}) \qquad \forall i \in O \tag{15}$$

Cut set (16)

(17)

$$u_{i,k}, u'_{i,k} \in [0,1], a_{i,s} \in \{0,1\}; q_{i,k}, T_i \in R^+ \qquad \forall i \in O, k \in K, s \in S \tag{18}$$

Algorithm 1 ASSIGN

Input: Model GOAS_RTBM, $p'_i, i \in O$
if First iteration of the GOAS_RTBM **then**
    Solve model GOAS_RTBM (p')
else
    Solve model GOAS_RTBM (p') with cuts derived by SCHEDULE
end if
Output: $a_{i,s}$, $i \in O, s \in S$

- ASSIGN gives a new assignment of orders to time segments in each iteration
- SCHEDULE handles each time segment separately (i.e. $|S|$ subproblems are solved sequentially by the MILP model)
- O_s denotes the set of orders assigned to time segment s
- Each time segment $s \in S$ has a beginning and end point. Thus, release times and deadlines of the orders in set O_s are modified as:

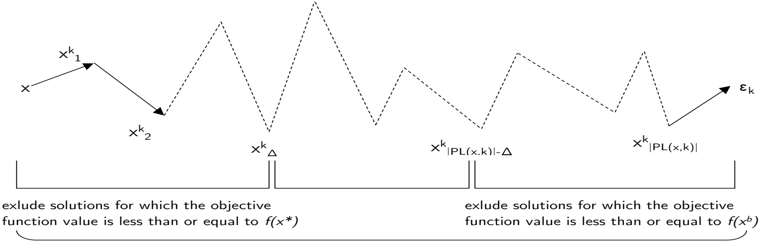
$$r_i^s = \max\{\varsigma_{s-1}, r_i\} \qquad \forall i \in O, s \in S$$
$$\bar{d}_i^s = \min\{\varsigma_s, \bar{d}_i\} \qquad \forall i \in O, s \in S$$

- $MILP(O_s, r^s, \bar{d}^s)$: The MILP model considering the orders in set O_s with their modified r and $\bar{d}$ parameters

Algorithm 2 SCHEDULE

Input: $O_s, r^s, \bar{d}^s, s \in S$
Initialize: $s = 1, x_0 = \emptyset$
while $s \leq |S|$ **do**
  $\pi_s \leftarrow MILP(O_s, r^s, \bar{d}^s)$
  $x_s \leftarrow x_{s-1} +\!\!+ \pi_s$
  if $s < |S|$ **then**
    Create x_s' by adding rejected orders so far to the partial solution x_s
    $x_s \leftarrow VNS(x_s')$
  end if
  $s + +$
end while
Output: $x_{|S|}$

- Short term memory to avoid reversal of moves and cycling: Recency based tabu list
- Local search algorithm / Move: Variable neighborhood descent using swap and insertion moves
- Evaluation of a move: Objective value and makespan
- Tabu move: An order that is swapped or inserted is tabu
- Tabu Tenure: 5 iterations
- Aspiration criteria: Best objective value
- Generation of initial solution for the new start: Via path relinking



- ASSIGN sets assignment of orders to time segments
- SCHEDULE
  - generates cuts to avoid ASSIGN generating same assignment of orders
    - If there is at least one $s \in S : O_s \not\equiv A_s$ where A_s is the set of accepted orders in set O_s

$$\sum_{i \in A_s} a_{i,s} |O_s \setminus A_s| + \sum_{i \in O_s \setminus A_s} a_{i,s} \leq |A_s||O_s \setminus A_s| \qquad \forall s \in S : O_s \not\equiv A_s.$$

    - else

$$\sum_{s \in S, i \in O_s} a_{i,s} \leq \sum_{s \in S} |O_s| - 1.$$

  - modifies parameter p' (input of ASSIGN):
    - st'_i: Estimation of the setup time before order i in the optimal solution (initially, it is set to the minimum setup time that can be observed before order i, $\forall i \in O$)
    - $st'_i = 0.9 st'_i + 0.1 s^x_i$ where s^x_i denotes the setup time executed before order i in solution x
    - $p'_i = p_i + st'_i$, $\forall i \in O$.

- Computational performance of GOAS_I2PM is compared with that of both the proposed mathematical model and the state-of-the-art algorithms SSP (Silva et al., 2018), SPARROW (He et al., 2019), FPTAS and BALAS (de Weerdt et al., 2021) for the GOAS problem.
- Instances generated by Cesaret et al. (2012) are used as previous studies.
- $\text{Gap} = \dfrac{UB_{instance} - LB_{instance}^{method}}{UB_{instance}}$

Computational results

Comparison of the proposed MILP and SSP for small size instances

n	Gap (%) SSP Min	Gap (%) SSP Avg	Gap (%) SSP Max	Gap (%) MILP Min	Gap (%) MILP Avg	Gap (%) MILP Max	NO	NF SSP	NF MILP	CPU time (s) SSP	CPU time (s) MILP
10	0.000	0.000	0.000	0.000	0.000	0.000	250	250	250	0.41	0.13
15	0.000	0.001	0.007	0.000	0.000	0.000	250	249	250	98.84	0.73
20	0.000	0.001	0.011	0.000	0.000	0.000	250	249	250	360.02	5.07
25	0.000	0.004	0.043	0.000	0.000	0.000	250	248	250	762.91	158.84

Comparison of GOAS_I2PM with the proposed MILP for $n = 50$

n	τ	R	Gap (%) MILP Min	Avg	Max	GOAS_I2PM Min	Avg	Max	NO	NF MILP	NF GOAS_I2PM	Solution time (s) MILP	Solution time (s) GOAS_I2PM
50	0.1	0.1	0.00	0.00	0.00	0.00	0.02	0.19	10.00	10.00	9.00	145.04	900.00
		0.3	0.00	0.00	0.00	0.00	0.02	0.17	10.00	10.00	9.00	166.12	900.00
		0.5	0.00	0.36	2.03	0.00	0.00	0.00	10.00	5.00	10.00	2762.04	900.00
		0.7	0.00	0.14	0.56	0.00	0.00	0.00	10.00	7.00	10.00	2372.16	900.00
		0.9	0.00	1.21	4.13	0.00	0.00	0.00	10.00	2.00	10.00	3568.20	900.00
	0.5	0.1	0.00	2.89	5.84	0.00	0.48	1.03	2.00	1.00	2.00	3600.00	900.00
		0.3	0.56	2.67	4.73	0.00	0.69	1.27	2.00	0.00	2.00	3600.00	900.00
		0.5	1.47	5.02	10.08	0.00	1.05	2.26	1.00	0.00	1.00	3600.00	900.00
		0.7	2.46	4.06	7.46	0.00	0.79	1.87	2.00	0.00	2.00	3600.00	900.00
		0.9	3.68	5.87	8.00	0.00	1.01	2.45	2.00	0.00	2.00	3600.00	900.00
	0.9	0.1	0.00	0.00	0.00	0.00	0.00	0.00	10.00	10.00	10.00	541.82	900.00
		0.3	0.00	0.07	0.65	0.00	0.02	0.24	10.00	9.00	9.00	1682.76	900.00
		0.5	0.00	0.54	2.15	0.00	0.00	0.00	10.00	6.00	10.00	3346.96	900.00
		0.7	0.00	0.30	0.84	0.00	0.10	0.56	10.00	3.00	8.00	3600.00	900.00
		0.9	0.00	0.85	2.47	0.00	0.00	0.00	10.00	3.00	10.00	3600.00	900.00
Avg:			0.50	1.80	3.53	0.00	0.57	1.31	109.00	66.00	104.00	2869.91	900.00

Computational results

Comparison of GOAS_I2PM with the proposed MILP for $n = 100$

n	τ	R	Gap (%) MILP Min	Gap (%) MILP Avg	Gap (%) MILP Max	Gap (%) GOAS_I2PM Min	Gap (%) GOAS_I2PM Avg	Gap (%) GOAS_I2PM Max	NO	NF MILP	NF GOAS_I2PM	Solution time (s) MILP	Solution time (s) GOAS_I2PM
100	0.1	0.1	0.00	0.00	0.00	0.00	0.04	0.26	10.00	10.00	8.00	1817.58	900.00
		0.3	0.00	0.45	1.86	0.00	0.07	0.19	3.00	1.00	3.00	3561.59	900.00
		0.5	2.44	6.91	11.04	0.00	0.00	0.00	10.00	0.00	10.00	3600.00	900.00
		0.7	5.75	9.34	12.95	0.00	0.00	0.00	10.00	0.00	10.00	3600.00	900.00
		0.9	12.34	16.27	25.68	0.00	0.00	0.00	10.00	0.00	10.00	3600.00	900.00
	0.5	0.1	17.43	23.38	36.90	0.34	0.52	0.88	0.00	0.00	0.00	3600.00	900.00
		0.3	18.03	23.94	30.46	0.33	0.68	0.96	0.00	0.00	0.00	3600.00	900.00
		0.5	18.70	31.26	45.03	0.28	0.56	0.80	0.00	0.00	0.00	3600.00	900.00
		0.7	27.06	34.75	44.86	0.00	0.51	0.99	1.00	0.00	1.00	3600.00	900.00
		0.9	20.26	30.53	42.12	0.00	0.34	1.47	4.00	0.00	4.00	3600.00	900.00
	0.9	0.1	0.58	1.34	2.13	0.00	0.22	0.52	10.00	0.00	2.00	3600.00	900.00
		0.3	0.79	2.18	2.92	0.00	0.17	0.40	10.00	0.00	3.00	3600.00	900.00
		0.5	1.42	6.23	10.02	0.10	2.83	5.68	5.00	0.00	0.00	3600.00	900.00
		0.7	8.53	11.67	14.83	4.74	6.99	10.34	0.00	0.00	0.00	3600.00	900.00
		0.9	9.81	14.68	21.46	2.68	6.64	10.27	0.00	0.00	0.00	3600.00	900.00
Avg:			10.70	16.22	23.03	0.53	1.23	2.08	73.00	11.00	51.00	3527.17	900.00

Computational results

Comparison of the number of optimal solutions found in the literature and in this study

		n=25		n=50		n=100	
τ	R	Literature	This study	Literature	This study	Literature	This study
0.1	0.1	3	10	0	10	0	10
	0.3	4	10	0	10	0	3
	0.5	7	10	3	10	6	10
	0.7	10	10	10	10	10	10
	0.9	10	10	10	10	10	10
0.3	0.1	6	10	0	9	0	0
	0.3	3	10	0	9	0	0
	0.5	3	10	1	8	0	0
	0.7	9	10	6	9	5	7
	0.9	8	10	8	9	8	10
0.5	0.1	10	10	0	2	0	0
	0.3	8	10	1	2	0	0
	0.5	8	10	0	1	0	0
	0.7	9	10	0	2	0	1
	0.9	9	10	1	2	1	4
0.7	0.1	10	10	0	0	0	0
	0.3	10	10	1	0	0	0
	0.5	10	10	1	0	0	0
	0.7	10	10	7	3	0	0
	0.9	10	10	7	4	0	0
0.9	0.1	10	10	10	10	10	2
	0.3	10	10	10	10	10	4
	0.5	10	10	10	10	5	0
	0.7	10	10	10	10	0	0
	0.9	10	10	10	10	0	0
Sum:		207	250	106	160	65	71

			Gap (%)															CPU time (s)				
n	τ	R	FPTAS ($\epsilon = 0.1$)			FPTAS ($\epsilon = 0.05$)			Balas (5)			Balas (12)			GOAS_I2PM			FPTAS ($\epsilon = 0.1$)	FPTAS ($\epsilon = 0.05$)	BALAS (5)	BALAS (12)	GOAS_I2PM
			Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max					
50	0.9	0.1	0.42	2.40	4.21	0.40	1.12	1.95	0.00	0.85	2.22	0.00	0.00	0.00	0.00	0.00	0.00	0.96	1.09	0.27	0.48	900.00
		0.3	0.90	2.64	6.05	0.45	1.23	2.19	0.48	3.00	6.52	0.00	0.11	1.09	0.00	0.02	0.24	1.87	2.15	0.30	1.05	900.00
		0.5	0.92	2.39	3.61	0.23	1.29	3.04	0.60	5.76	9.37	0.00	0.56	1.49	0.00	0.00	0.00	9.04	11.00	0.59	5.91	900.00
		0.7	1.80	2.88	4.30	0.55	1.22	1.85	2.05	6.30	10.70	0.08	1.19	2.99	0.00	0.10	0.56	30.00	33.00	0.81	7.50	900.00
		0.9	0.35	2.44	4.10	0.90	1.33	2.11	3.33	6.28	12.30	0.00	1.33	3.50	0.00	0.00	0.00	269.00	325.00	1.20	29.30	900.00
100	0.9	0.1	0.63	1.82	3.27	0.32	0.92	1.80	1.65	2.50	3.45	0.00	0.15	0.40	0.00	0.22	0.52	358.00	453.00	10.10	134.00	900.00
		0.3	1.42	1.94	-	0.53	0.80	1.31	4.98	6.73	9.47	0.31	1.49	2.25	0.00	0.17	0.40	1230.00	1130.00	4.72	104.00	900.00
		0.5	0.88	1.61	-	0.73	0.96	-	4.97	6.73	9.99	0.79	2.12	3.77	0.10	0.40	0.75	1690.00	2000.00	5.39	200.00	900.00

Computational results

Comparison of GOAS_I2PM with the state-of-the-art algorithms SSP and SPARROW for $n = 50$

n	τ	R	Gap (%) SSP Min	SSP Avg	SSP Max	SPARROW* Min	SPARROW* Avg	SPARROW* Max	GOAS_I2PM Min	GOAS_I2PM Avg	GOAS_I2PM Max	CPU time (s) SSP	CPU time (s) GOAS_I2PM
50	0.1	0.1	1.20	2.24	3.00	0.51	0.79	1.27	0.00	0.29	0.60	3415.80	900
		0.3	1.39	2.44	3.92	0.60	1.00	1.53	0.30	0.53	0.78	3600.00	900
		0.5	1.12	1.61	2.10	0.00	0.42	0.96	0.00	0.04	0.42	3217.10	900
		0.7	0.00	2.45	16.39	0.00	1.66	16.38	0.00	1.64	16.39	2661.80	900
		0.9	0.00	0.54	1.90	0.00	0.00	0.00	0.00	0.00	0.00	2139.00	900
	0.3	0.1	0.52	2.35	3.48	0.63	1.59	2.86	0.41	1.02	1.72	3600.00	900
		0.3	2.66	3.54	5.05	1.40	1.81	2.69	0.77	1.28	2.15	3600.00	900
		0.5	1.33	3.49	11.87	0.18	1.44	3.64	0.00	1.01	3.10	3600.00	900
		0.7	0.00	1.40	3.44	0.00	0.39	1.16	0.00	0.07	0.35	3312.10	900
		0.9	0.40	1.35	2.85	0.00	0.27	1.32	0.00	0.06	0.61	3250.10	900
	0.5	0.1	0.57	4.32	13.76	1.11	2.01	2.64	0.56	1.42	2.15	3558.70	900
		0.3	2.37	4.83	7.80	1.97	3.30	5.44	1.64	2.68	4.77	3017.60	900
		0.5	1.55	3.84	8.01	1.03	3.00	6.49	0.34	2.41	5.52	3600.00	900
		0.7	1.51	2.91	5.16	0.37	1.97	4.14	0.00	1.43	3.31	3600.00	900
		0.9	0.35	2.25	4.20	0.00	1.94	4.04	0.00	1.31	3.64	3484.60	900
	0.7	0.1	2.42	3.93	5.20	2.24	3.88	4.74	1.67	3.44	4.45	3600.00	900
		0.3	2.50	4.27	7.60	2.75	4.82	8.40	2.48	4.17	7.42	3160.50	900
		0.5	4.55	6.16	10.32	4.74	6.46	10.91	4.55	6.07	10.32	3356.50	900
		0.7	1.03	6.55	14.31	1.73	7.13	15.50	1.19	6.56	14.31	1971.70	900
		0.9	2.52	7.16	13.00	3.26	7.63	13.02	2.52	7.16	13.00	1374.30	900
	0.9	0.1	6.55	10.88	16.77	6.55	10.98	16.77	6.55	10.88	16.77	304.60	900
		0.3	9.28	13.25	17.40	9.28	13.43	17.80	9.28	13.27	17.60	15.70	900
		0.5	2.63	12.15	16.52	3.05	12.31	16.78	2.63	12.15	16.52	76.10	900
		0.7	5.62	11.13	17.64	5.62	11.50	18.42	5.62	11.21	17.64	54.00	900
		0.9	9.26	12.09	15.85	9.74	12.49	16.30	9.26	12.09	15.85	238.00	900
Avg:			2.45	5.09	9.10	2.27	4.49	7.73	1.99	4.09	7.18	2552.33	900

* He et al. (2019) do not share the run time of SPARROW, just state that it is under a minute on average for $n = 50$.

Computational results

Comparison of GOAS_I2PM with the state-of-the-art algorithms SSP and SPARROW for $n = 100$

n	τ	R	Gap (%) SSP Min	SSP Avg	SSP Max	SPARROW* Min	SPARROW* Avg	SPARROW* Max	GOAS_I2PM Min	GOAS_I2PM Avg	GOAS_I2PM Max	CPU time (s) SSP	CPU time (s) GOAS_I2PM
100	0.1	0.1	1.18	2.44	3.28	0.39	0.57	0.80	0.09	0.24	0.43	3600.00	900
		0.3	1.51	2.10	3.06	0.34	0.61	0.88	0.09	0.26	0.53	3600.00	900
		0.5	0.36	1.39	2.74	0.00	0.13	0.28	0.00	0.00	0.00	3423.00	900
		0.7	0.00	0.43	0.80	0.00	0.00	0.00	0.00	0.00	0.00	2243.60	900
		0.9	0.00	0.22	0.89	0.00	0.00	0.00	0.00	0.00	0.00	2354.80	900
	0.3	0.1	1.47	2.65	3.96	0.53	0.99	1.44	0.20	0.58	1.26	3600.00	900
		0.3	1.94	2.94	5.39	0.57	1.24	2.49	0.37	0.77	1.93	3600.00	900
		0.5	1.14	2.21	3.81	0.65	1.07	1.75	0.36	0.62	1.02	3600.00	900
		0.7	0.37	1.65	2.75	0.00	0.30	0.79	0.00	0.08	0.36	3515.40	900
		0.9	0.19	0.91	2.41	0.00	0.11	0.42	0.00	0.00	0.00	2788.30	900
	0.5	0.1	2.49	3.96	5.27	1.07	1.79	2.71	0.49	1.23	2.12	3600.00	900
		0.3	2.68	3.95	5.99	1.49	2.16	2.63	0.86	1.57	2.09	3600.00	900
		0.5	2.57	3.54	4.53	1.31	2.33	3.35	0.70	1.69	2.66	3600.00	900
		0.7	1.61	2.71	4.04	0.35	1.49	2.38	0.00	0.82	1.80	3600.00	900
		0.9	0.67	2.10	4.65	0.19	0.92	2.40	0.00	0.41	1.79	3600.00	900
	0.7	0.1	2.70	4.98	6.42	1.69	2.47	3.21	1.27	1.95	2.74	3600.00	900
		0.3	3.93	6.66	10.83	1.35	3.66	5.80	0.97	3.04	5.17	3600.00	900
		0.5	4.04	6.45	12.85	2.23	4.34	10.65	1.60	3.78	10.03	3600.00	900
		0.7	3.36	7.36	13.20	1.92	4.81	7.33	1.60	4.28	6.72	3600.00	900
		0.9	5.30	8.40	12.95	2.97	5.07	6.55	2.62	4.52	6.05	3600.00	900
	0.9	0.1	4.59	8.70	11.95	3.44	5.46	8.47	3.45	5.27	8.29	3600.00	900
		0.3	5.68	12.45	17.41	4.90	8.65	11.20	4.53	8.25	10.77	3600.00	900
		0.5	10.80	15.15	18.38	7.90	10.76	15.67	7.48	10.58	15.52	3600.00	900
		0.7	10.70	14.97	19.95	7.38	10.73	13.02	7.03	10.53	12.57	3600.00	900
		0.9	11.23	14.95	20.47	3.32	10.67	16.53	3.36	10.54	16.32	3600.00	900
Avg:			3.22	5.33	7.92	1.76	3.21	4.83	1.48	2.84	4.41	3453.00	900

* He et al. (2019) do not share the run time of SPARROW, just state that it is under a minute on average for $n = 100$.

Conclusion

- New MILP model for the GOAS problem
- A new relaxed time-bucket formulation
- A generic matheuristic approach based on parameter relaxation (relaxation of setup times in our problem)
- A multi-start granular tabu search algorithm with a new diversification approach
- Future research:
  - New relaxed models including setup times for ASSIGN
  - Branch and bound algorithm through assignment variables