Casual Employee Scheduling with CP, ACO and VND

EUROCAST 2019, Las Palmas de Gran Canaria, Spain

Nikolaus Frohner, Stephan Teuschl, and Günther R. Raidl

February 18, 2019

ALGORITHMS AND COMPLEXITY GROUP

Casual Employee Scheduling

We consider a real-world employee scheduling problem of an Austrian association, where **casual employees** (workers) have to be **assigned** to **shifts** to satisfy **requirements** of different work **locations**.

Goal: for a given planning horizon, find an assignment that satisfies a number of **hard constraints** while minimizing an objective function consisting of **soft constraints**.

Example Assignment

Table: Left: workers, center: requirements, right: assignments

w	N_w	L_w
0	3	$\{2\}$
1	2	$\{0,1,2\}$
2	1	$\{0\}$

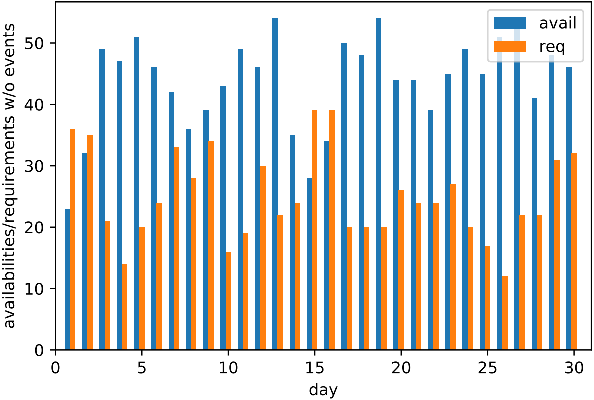
$R_{(l,d)}$	0	1	2
0	1	1	1
1	0	1	1
2	2	1	0

$A_{(l,d)}$	0	1	2
0	$\{0\}$	$\{2\}$	$\{1\}$
1	$\{\}$	$\{1\}$	$\{0\}$
2	$\{\}$	$\{0\}$	$\{\}$

Small example of an assignment with three locations, three days and three locations, availabilities omitted.

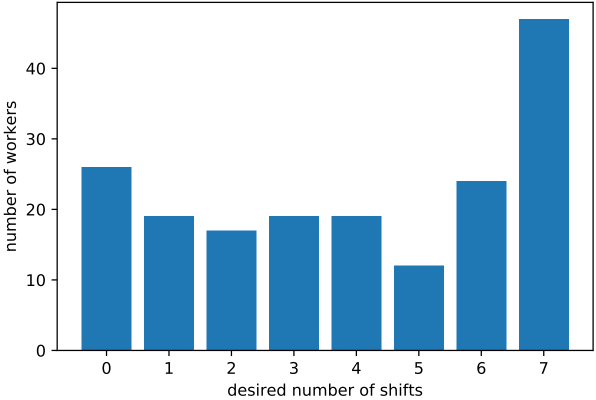
$N_w \ldots$ worker's desired number of shifts.
$L_w \ldots$ preferred locations.
$R_{(l,d)} \ldots$ number of workers required by shift $s = (l, d)$.
$A_{(l,d)} \ldots$ solution, actual assignment of workers to shifts.

Availabilities and Requirements within Month



Casual: employees place offers for shifts they are available, varying over time, instead of having a contract with a fixed number of hours per month.

Desired Number of Assigned Shifts



Employees have a desired number of shifts. This desired number is also an “at most” hard constraint.

Hard Constraints

Amongst others:

1. Every available employee has to get a least one shift and at most the desired number of shifts.
2. Special shifts have to be all satisfied:
    2.1 **Floating**: employee will be assigned to a concrete location on the very day of the shift to account for prediction uncertainty of actual requirement.
    2.2 **Standby**: employee will only work when someone gets sick.
3. Employee only eligible for standby shifts when assigned to enough regular shifts.

Shift Shortage

Since usually shortage of employees, no hard constraint concerning requirement satisfaction—instead modeled as mean squared error (MSE) penalty term w.r.t. how short a shift is of required staff.

$$g_u(A) := \frac{1}{|S|} \sum_{s \in S} \left(\underbrace{1 - \frac{|\{A_w | s \cap A_w \neq \emptyset\}|}{R_s}}_{\text{shift shortage}} \right)^2$$

$s \ldots$ shift, the tuple location and day.
$A_w \ldots$ assignment of worker w, a set of shifts.
$R_s \ldots$ number of workers required by shift s.

Fairness Constraints

MSE also models fairness of shortage across shifts. Employees also deserve fairness regarding:

1. Number of assigned shifts normalized by number of shifts desired to work should be as equal as possible; modeled as variance.
2. The same separately for floating shifts, since they are annoying.
3. Fraction of shifts they work in **not preferred locations** should be as small and as equal as possible; modeled as MSE.

Multi-objective Optimization Problem (MOOP)

We ended up with a MOOP. We convert into a SOOP by means of a weighted sum approach and put the most focus on reducing the shift shortage—may be tuned further when in productive use.

$$f(A) := \lambda_u g_u(A) + \lambda_f g_f(A) + \lambda_{ff} g_{ff}(A) + \lambda_{np} g_{np}(A)$$

For now $\boldsymbol{\lambda} = (10, 1, 1, 1)$.

Exact Approach

Nethercote et al. (2007)

- Constraint programming (CP) model in MiniZinc.
- Floats & Set formulation, decision variables: $assign[l, d] = W_{(l,d)}$. Supported by Gecode and JaCoP.
- variable ordering: first_fail, value ordering: indomain_min.

Hybrid Approach

Nethercote et al. (2007), Dorigo (1992), Mladenović and Hansen (1997)

- Feasible initial solution (assignment) generated by MiniZinc/CP Solver. Randomization by shuffling employees.
- To use powerful solver like Gurobi: Floats & 0-1 formulation, decision variables: $assign[l, d, w] \in \{0, 1\}$.
- Model-based randomized greedy completion of initial solution (Ant Colony Optimization (ACO)).
- Stochastic Variable Neighborhood Descent (VND).
- Elitist model-update (pheromones).
- Restart with randomization by shuffling employees.

Simpler Hybrid Approach as Baseline

CP solver gives us already solution that is not or little greedily extensible.

- Feasible initial solution (assignment) generated by MiniZinc/CP Solver.
- Stochastic VND including shift assignment neighborhood.
- Restart with randomization by shuffling employees.

Solution Construction by Ants

Given a current assignment A. Let $S_u(A) \ni s$ be the shifts with shortage and workers $W_s \ni w_s$ that are available for this shift and fulfill all the hard constraints.

Select $s \leftrightarrow w_s$ to create A' with probability, until no more feasible extension possible:

$$p_{s,w_s} \sim \frac{1}{(f(A') - f(A))^{\alpha}} \tau_{s,w_s}^{\beta}$$

Neighborhood Structures

1. Assign unassigned shift to worker.
2. Reassign shift from one worker to another.
3. Reassign worker from one shift to another.
4. Swap shift between workers.
5. Change location of day for worker.

Iterative Stochastic VND

- VND with random step function.
- Search exhaustiveness parameter e: number of neighbors without improvement until switching to next neighboorhood.

Pheromone Model

Stützle, Hoos (2000)

Min-Max ACO with $[\tau_{\text{min}}, \tau_{\text{max}}]$.

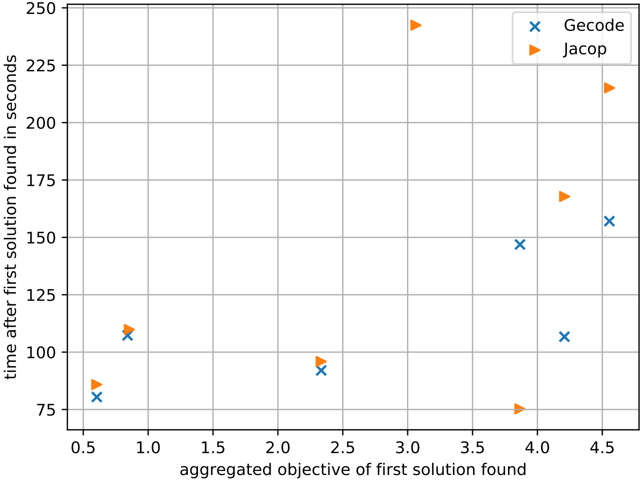
Best solution found so far increase pheromones τ_{s,w_s} by $1/f(A_{bs})$.

Evaporation controlled by parameter ρ.

Artificial Instances

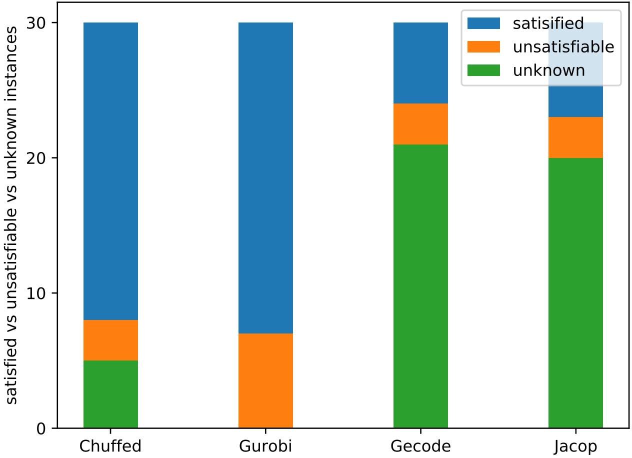
Generated 30 artificial instances randomly. Used for CP solver selection and tuning algorithmic parameters.

First Solution Gecode vs Jacop



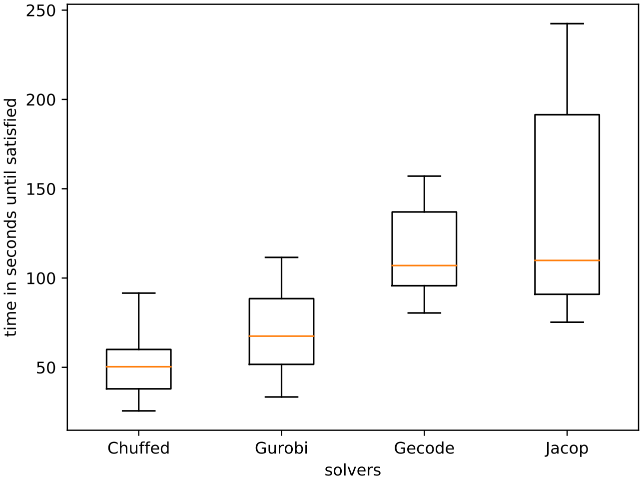
Solver comparison with artificial instances in optimization mode.

Solvers Comparison I



Statistics after one hour regarding satisfaction.

Solvers Comparison II



Time in seconds until satisfied, only for satisfiable instances.

Preliminary Results Real-world Instances

Four precious real world instances.

Experiments with a time limit of one hour on Intel Xeon E5-2640 with 16GB memory (32GB for October instance).

instance	$\|W\|$	$\sum_s R_s$	$\|D\|$	UB_c	Solver	t_f $[s]$	Exact	CSP	+VND
Sep 2018#1	183	930	30	0.903	JaCoP	132	1.846	1.951	0.520
Sep 2018#2	184	928	30	0.888	Gecode	149	1.257	1.383	0.335
Oct 2018	253	1079	31	0.918	Gurobi	400	-	6.834	0.574
Feb 2019	172	685	28	1.001	JaCoP	108	0.245	0.255	0.193

UB_c . . . theoretical upper bound total shift coverage
t_f . . . time until first feasible solution found.

… more descriptive

More interesting are the components of the objective function.

instance	CSP+VND f	c	u	g_u	g_f	g_{ff}	g_{np}
Sep 2018#1	0.520	0.868	123	0.177^2	0.320^2	0.123^2	0.301^2
Sep 2018#2	0.335	0.877	114	0.135^2	0.249^2	0.142^2	0.267^2
Oct 2018	0.574	0.858	153	0.195^2	0.318^2	0.130^2	0.272^2
Feb 2019	0.173	0.988	8	0.022^2	0.307^2	0.130^2	0.277^2

For Sep 2018#2, there is also a manually created solution available:

instance	manual f	c	u	g_u	g_f	g_{ff}	g_{np}
Sep 2018#2	0.570	0.945	51	0.188^2	0.441^2	0.144^2	0.045^2

Violates our hard constraints, to be checked.

c … total coverage.

u … absolute amount of unfulfilled requirements.

Next Steps

1. Different variable and value ordering heuristics.
2. Ants.
3. Destroy & Recreate.
4. irace.
5. User acceptance test.

References

Nicholas Nethercote, Peter J Stuckey, Ralph Becket, Sebastian Brand, Gregory J Duck, and Guido Tack.
MiniZinc: Towards a standard CP modelling language.
In *International Conference on Principles and Practice of Constraint Programming*, pages 529–543. Springer, 2007.

Walter J Gutjahr and Marion S Rauner.
An ACO algorithm for a dynamic regional nurse-scheduling problem in austria.
Computers & Operations Research, 34(3):642–666, 2007.

F Bellanti, Giuliana Carello, Federico Della Croce, and Roberto Tadei.
A greedy-based neighborhood search approach to a nurse rostering problem.
European Journal of Operational Research, 153(1):28–40, 2004.

Thomas Stützle and Holger H Hoos.
MAX–MIN ant system.
Future generation computer systems, 16(8):889–914, 2000.

Nenad Mladenović and Pierre Hansen.
Variable neighborhood search.
Computers & operations research, 24(11):1097–1100, 1997.

Thanks for your attention.