Tabu search with guided restarts for a car production problem with a 2/3 balancing penalty

Jean Respen and Nicolas Zufferey

Geneva School of Economics and Management, University of Geneva, Switzerland
jean.respen@unige.ch n.zufferey@unige.ch

1 Presentation of problem (P)

Nowadays, new constraints known as *smoothing constraints* are attracting a growing attention in the area of job scheduling [1] and in particular for car sequencing problems, where cars must be scheduled before production in an order respecting various constraints (colors, optional equipment, due dates, etc.), while avoiding overloading some important resources. For the car plant, balancing between optional equipment and colors allows to respect customers deadlines and to prevent overloading some resources (machines or employees).

In 2005, the car manufacturer *Renault* proposes a car sequencing problem through the ROADEF 2005 Challenge [3], with real instances involving hundreds of cars. Car families are defined so that two cars of the same family contain the same optional equipment. Each optional equipment i is associated with a N_i/P_i *ratio constraint*, meaning that at most N_i cars with option i can be scheduled in any subsequence of P_i cars, otherwise a penalty occurs. The objective is to minimize a weighted function involving ratio constraint violations and the number of color changes. In [2], a variation of the *Renault* problem is studied, where non-identical parallel machines (or production lines) and eligibility constraints are considered (i.e. a job – or a car – can only be performed on some specific machines). The objective function involves three components related to makespan, smoothing costs and setup costs. In this work, we study another variant of the car sequencing problem, where the violations of the 2/3 ratio constraint are penalized as smoothing costs in the objective function, with eligibility and makespan constraints.

In the considered problem (P) are considered n jobs, m non-identical machines and eligibility constraints. Each job j belongs to one of the g available families and has a processing time p_{ij} depending on the machine i. A solution s contains a production sequence for each machine. The goal consists in minimizing the smoothing cost function $f(s)$, which is the weighted number of times we have three consecutive jobs of the same family in s. In addition, the overall makespan cannot exceed an upper bound UB (but UB is set large enough to easily prevent the rejection of jobs). A small value of UB usually indicates a high occupancy rate of the machines, and as a consequence, the production system will be available sooner for future commands.

2 Solution methods for (P)

Three different methods are proposed: GR (a greedy heuristic), TS (a conventional tabu search), and $TSGR$ (a tabu search with *guided* restarts, managed with a distance function). The time limit of each algorithm is $T = 15$ minutes (which is consistent from a practical standpoint). Note that if an algorithm stops before T, it is restarted and the best generated solution is returned to the user.

GR starts from an empty solution s. At each step, it inserts the job j in s which minimizes the augmentation of f, while respecting the eligibility and makespan constraints. Each possible insertion is tested and ties are broken randomly. GR stops when all jobs are scheduled.

TS starts from an initial solution provided by GR and, at each iteration, tries to improve it by performing the best possible non tabu move. A *move* is defined as positioning a job somewhere else in the solution (in the same sequence or in the sequence of another machine). Each time a move is performed, it is forbidden (tabu) to move it again for *tab* iterations, where *tab* is uniformly generated in interval $[3, 7]$ after each move.

In $TSGR$, guided restarts of TS are performed as follows, where a *cycle* is defined as an execution of TS for $I = 100$ iterations. Let $s_b^{(k)}$ (resp. $s_i^{(k)}$) be the best visited (resp. initial) solution in cycle k. The *distance* between two solutions s_1 and s_2 is defined as $dist(s_1, s_2) = \sum_j y_j(s_1, s_2)$, where $y_j(s_1, s_2) = 1$ if job j is has the same position index in solutions s_1 and s_2 (for the sequence it belongs to, independently of the machine), and $y_(s_1, s_2) = 0$ otherwise. Note that the same sequence of jobs can appear on two different machines for s_1 and s_2, which is consistently measured as equivalent situations by the distance function. Then, at the end of a cycle k, if $dist[s_b^{(k-1)}, s_b^{(k)}] < n/4$, $s_i^{(k+1)}$ is generated by performing 10 random swap moves on $s_b^{(k)}$, otherwise $s_i^{(k+1)}$ is generated with GR. Note that *swap* moves are defined as exchanging the position index of two jobs on the same machine. This mechanism allows to intensify the search if the two best solutions of two consecutive cycles have a similar structure. Otherwise a diversification action is triggered with a restart.

3 Results

An exact linear formulation relying on CPLEX 12.4 has been tested with a time limit of 10 hours on an Intel Quad-core i7 @ 3.4 GHz with 8 GB DDR3 of RAM memory. CPLEX is only able to solve instances with up to 30 jobs, for which the proposed tabu search approaches are usually able to quickly find optimal solutions. For these reasons, exact methods will not be discussed further.

The instances are derived from the ones presented in [2]. Methods GR, TS and $TSGR$ are compared in Table 1. For each instance are first given n, m, UB and $f^\star$, which is the best solution value found by any of the algorithm. The next column indicates the percentage gap between $f^\star$ and the best solution value found by GR within $T = 15$ minutes. The last two columns present the same information for TS and $TSGR$ (but the results are averages over 10 runs with $T = 15$ minutes). The last row indicates the average gaps for the three methods. The best result for each instance is highlighted in bold face. It can be observed that: (1) TS is much more efficient than GR, which shows the relevance of the used moves; (2) $TSGR$ significantly outperforms TS, which indicates that the proposed way to guide the restarts is powerful, and should be investigated for other problems.

Table 1. Results on instances with 100 and 300 cars

n	m	UB	$f^\star$	GR	TS	$TSGR$
100	4	3604	3005	4.49%	5.39%	**0.00**%
100	4	3612	3005	4.49%	2.70%	**0.00**%
100	4	3610	2980	5.37%	0.39%	**0.00**%
100	4	3632	2850	10.18%	4.02%	**0.70**%
300	5	9005	960	42.92%	8.72%	**2.08**%
300	5	9038	895	47.82%	7.31%	**4.02**%
300	5	9086	870	52.07%	5.32%	**1.38**%
300	5	9143	730	81.23%	12.77%	**0.68**%
Average				31.07%	5.83%	**1.11**%

References

1. C. Becker and A.Scholl. A survey on problems and methods in generalized assembly line balancing. European Journal of Operational Research, 168(3):694 - 715, 2006.
2. J. Respen, N. Zufferey, and E. Amaldi. Heuristics for a multi-machine multi-objective job scheduling problem with smoothing costs. In 1st IEEE International Conference on Logistics Operations Management 2012, 2012.
3. C. Solnon, V. Cung, A. Nguyen, and C. Artigues. The car sequencing problem: Overview of state-of-the-art methods and industrial case-study of the ROADEF 2005 challenge problem. European Journal of Operational Research, 191(3):912-927, 2008.