Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

MARTIN GRÖTSCHEL SVEN O. KRUMKE JÖRG RAMBAU
LUIS M. TORRES

Making the Yellow Angels Fly: Online Dispatching Of Service Vehicles in Real Time

The forth author was supported by DAAD

ZIB-Report 02-18 (March 2002)

MAKING THE YELLOW ANGELS FLY: ONLINE DISPATCHING OF SERVICE VEHICLES IN REAL TIME

MARTIN GRÖTSCHEL, SVEN O. KRUMKE, JÖRG RAMBAU, AND LUIS M. TORRES

ABSTRACT. Combinatorial online optimization is an area with lots of applications and potential for significant progress, both in theory and practice. In this short note we sketch the ADAC problem, a typical large-scale online optimization problem, discuss some theoretical and practical issues coming up, and explain, very briefly, how we approach this problem mathematically. Online problems are a battlefield of heuristics with many strong claims about their solution quality. We indicate that a stronger problem orientation and the use of a little more mathematics may yield.

1. THE PROBLEM

The German automobile club *ADAC* (*Allgemeiner Deutscher Automobil-Club*), second in size only to the AAA, the largest organization in the world, maintains a heterogeneous fleet of over 1,600 service vehicles in order to help people whose cars break down on the road. All ADAC service vehicles have a distinctive yellow color, and the help of provided by the experienced mechanics often appears to have been sent from heaven; for these reasons, the service vehicles are affectionally called "yellow angels." In this article we are a little more down-to-earth and just refer to them as "units."

ADAC runs five help centers (*Pannenhilfezentralen*), which are located throughout Germany and operate twenty-four hours, seven days a week. (Human) dispatchers have to reply to help requests (*events*, for short) instantly. Their task is to assign a unit (equipped with GPS) to serve each customer and to predict the estimated time of arrival at the customer's location.

In addition to the ADAC fleet, about 5,000 units operated by service contractors can be employed to cover events that otherwise could not be served in time. In the continuously running planning process, the dispatcher can revoke a particular assignment to an ADAC unit at a later moment in time, provided that the unit has not reached the customer.

There is no unique objective. The goals are high-quality service (e.g., short waiting times for the customers, preferably well below one hour but depending on the system load) and low operational costs (through measures like short total tour lengths and limited overtime expenditures). Figure 1 shows a typical, obviously not uniform distribution of units (gray) and events (black) in a real world instance of ADAC.

With increasing costs and request volume, ADAC's dispatching system has come under stress. The challenge is to design an automatic online dispatching

The forth author was supported by DAAD.

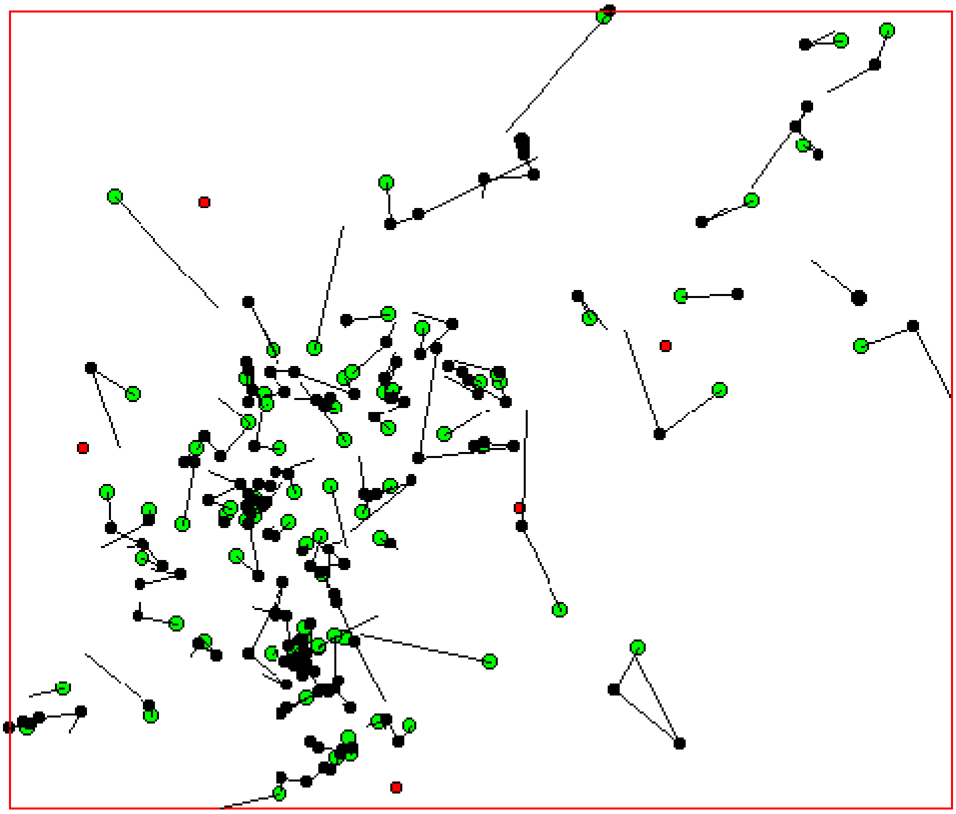

FIGURE 1. Partial snapshot of a real-world instance of the ADAC-problem (gray: units, black: events). Lines indicate currently planned tours

system that will guarantee small waiting times for events and will keep operational costs low. The modeling of such a task is not easy, as many technical and organizational side constraints—some hard, others soft—have to be considered, and it takes some time to figure out which constraints and objectives really count. We do not describe the complete model here but consider, as one example of a management decision, ADAC's imposition of a soft deadline on the service time of an event: The deadline may be missed at the cost of a linearly increasing lateness penalty (*soft time windows*).

In the remainder of this article we refer to the dispatching problem sketched above as the *ADAC-problem*.

2. THE PROBLEM WITH THE PROBLEM

In fact, we are facing *two* problems. The first is the task to compute an optimal dispatch for a "snapshot" situation (that is, a dispatch which is optimal for a given set of events and contractors at some moment in time). Because it contains the classic vehicle-routing problem as a special case, this problem is NP-hard. The difficulty is that in the dynamic planning process, a dispatch now has to be returned in no more than 15 seconds (*realtime-problem*).

A problem of this type can be described as a multidepot vehicle-routing problem with *soft* time windows. To date, most research in that area has considered *hard* time windows (see [1] and references therein for a survey of various problem types and exact algorithms; see [2] for recent progress in efficiency of exact algorithms). For instant answers, the literature seems to suggest that use of meta-heuristics is unavoidable ([4] describes a tabu search approach dealing with soft time windows; see [3] presents an approach based on genetic algorithms). These meta-heuristics, however, do not provide evaluations of their results, a significant draw-back in our opinion.

Integer programming, the approach we prefer (maybe in conjunction with whatever heuristics turn out to be useful), often yields *lower bounds* on the costs that are unavoidable with any solution method. It was this lower-bound-argument that convinced the ADAC not to simply pay one of the many commercial providers of meta-heuristic optimization, but to let us look a little deeper into the problem.

The second problem we face is more subtle: We do not know where future events will pop up (*online-problem*). A decision that is "optimal" at some point in time can prove later to have been unwise. In particular, even if we were able to compute locally optimal dispatches for any snapshot situation this does *not* mean that we obtain a dispatch which is (in hindsight) optimal for the whole planning period.

3. Towards a Solution

A closer inspection of the problem and typical instances leads us to expect that "snapshot optimization" does have potential. Time windows are helpful. Although they often seem to make routing problems harder, the contrary is true in our online situation. We compute optimal snapshot-dispatches whenever new events occur and use those results for the subsequent routing decisions until the arrival of new events. In planning tours for the service units, we have each unit end its tour at its home base; in this way we avoid units' drifting far from home and requiring expensive return trips at the end of their shifts.

For the real-time problem, the ADAC-problem luckily turns out to have some structure that we can use to our advantage. Because it takes some time for a service unit to deal with an event and because the time windows for events are quite tight, tours in an optimal solution cannot be too long (at most five events per unit). So, again because of the time windows, chances are good that all relevant tours can be enumerated implicitly.

What comes to mind at this point is a well-known column-generation and set-partitioning approach: Generate all relevant tours and partition the set of events into a subset of tours, with some single events outsourced to the contractors. The fractional set-partitioning solution (LP-relaxation) serves as a lower bound (unavoidable costs) for a snapshot-dispatch. This is basically what is done in most of the known exact algorithms in vehicle routing. Our task was to turn the known methods—which usually require substantial time for warmup or even preprocessing—into an algorithm that delivers provably good snapshot-dispatches within seconds.

We accomplish this with a *dynamic pricing control* in the column generation procedure. Early in the optimization, we work on an extremely restricted search space for new tours, using a mix of heuristics to find new tours of restricted depth. The search space for new tours is extended as the estimates for the dual prices in the LP relaxation of the set partitioning problem get closer to the truth.

4. Results

Our implementation of the column generation method ZIBDIP (ZIB dispatching via integer programming) was competing with an experimental meta-heuristic prototype programmed for this particular purpose by ADAC's software provider, Intergraph Public Safety (IPS). The evaluation was done on varying amounts of snapshot data. The results were conclusive: In this special application, the integer programming approach completely outperforms the meta-heuristic approach. ZIBDIP produced optimal(or optimal to within 1%) solutions for all real-world examples (up to 215 events for 98 units) within one to twenty seconds on a standard PC with 800 MHz; the meta-heuristic, by contrast, was unable in several instances to get closer to optimality than 25%. In larger, artificially generated problems (up to 775 events for 215 units), this trend was vastly amplified.

The superb performance on real data convinced ADAC and IPS that our algorithm was the way to go for the snapshot-dispatches, and IPS agreed on implementing the production software based on the ZIBDIP algorithm.

With the realtime issue solved to everyone's satisfaction, we turned our attention to the online issue. What quality would the repeated generation of optimal snapshot-dispatches deliver over time?

Although the development of the final version of the online algorithm is still under way, we have run online tests comparing two strategies: planning snapshot-dispatches after each new job or at fixed frequencies (one to five minutes). The trend was slightly in favor of planning after each new job.

The results were promising: On average, the online costs ranged 40% above the lower bound—not too bad, in our experience, for an online algorithm. We hope, however, to improve on these figures by utilizing estimates of future events in the snapshot-dispatches. This is work in progress and leads to most interesting questions as to how knowledge about the future can be exploited in combinatorial online optimization.

References

1. Jaques Desrosiers, Yvan Dumas, Marius M. Solomon, and François Soumis, *Time constraint routing and scheduling*, Network Routing (Michael Ball, Tom Magnanti, Clyde Monma, and George Newhauser, eds.), Handbooks in Operations Research and Management Science, vol. 8, Elsevier, Amsterdam, 1995, pp. 35–140.
2. Jesper Larsen, *Vehicle routing with time windows—finding optimal solutions efficiently*, DORSnyt (engl.) (1999), no. 116.
3. Sushil J. Louis, Xiangying Yin, and Zhen Ya Yuan, *Multiple vehicle routing with time windows using genetic algorithms*, 1999 Congress on Evolutionary Computation (Piscataway, NJ), IEEE Service Center, 1999, pp. 1804–1808.
4. Éric Taillard, P. Badeau, Michel Gendreau, F. Guertin, and Jean-Yves Potvin, *A tabu search heuristic for the vehicle routing problem with soft time windows*, Transportation Science **31** (1997), 170–186.

MARTIN GRÖTSCHEL, ZUSE-INSTITUT BERLIN, TAKUSTR. 7, 14195 BERLIN, GERMANY
E-mail address: rambau@zib.de

SVEN O. KRUMKE, ZUSE-INSTITUT BERLIN, TAKUSTR. 7, 14195 BERLIN, GERMANY

JÖRG RAMBAU, ZUSE-INSTITUT BERLIN, TAKUSTR. 7, 14195 BERLIN, GERMANY
E-mail address: rambau@zib.de

LUIS M. TORRES, ZUSE-INSTITUT BERLIN, TAKUSTR. 7, 14195 BERLIN, GERMANY
E-mail address: torres@zib.de