

www.palgrave-journals.com/jors/

The capacitated team orienteering and profitable tour problems

C Archetti[1]*, D Feillet[2], A Hertz[3] and MG Speranza[1]

[1]*University of Brescia, Brescia, Italy;* [2]*Université d'Avignon et des Pays de Vaucluse, Avignon, France; and* [3]*École Polytechnique and GERAD, Montréal, Canada*

In this paper, we study the capacitated team orienteering and profitable tour problems (CTOP and CPTP). The interest in these problems comes from recent developments in the use of the Internet for a better matching of demand and offer of transportation services. We propose exact and heuristic procedures for the CTOP and the CPTP. The computational results show that the heuristic procedures often find the optimal solution and in general cause very limited errors.

Journal of the Operational Research Society (2009) **60,** 831–842. doi:10.1057/palgrave.jors.2602603
Published online 28 May 2008

Keywords: team orienteering problem; profitable tour problem; column generation; branch and price; tabu search; variable neighbourhood search

1. Introduction

Carriers, as most of the other companies, nowadays meet the challenges of global competition. Markets are wider and wider, and often carriers need to look for customers far from their traditional service area because competitors have started serving that area. Competition also tends to force carriers to reduce their costs. This in turn results in an effort to use the capacity of the trucks at best. Seasonal effects often cause a non-optimal use of trucks, since, with a given fleet of vehicles, the capacity is fully exploited in some periods while it is not in other periods. Empty returns are an often experienced unpleasant fact. In Europe more than 30% of the trucks travel without any load and generate traffic and many more have a partial load. Carriers work hard to expand their market area, to attract new customers, to convince shippers that outsourcing the transportation function is convenient to them, to reduce costs by filling up the capacity of their trucks. One opportunity for carriers to reach these goals is to make use of the Internet. The Internet may encourage shippers to outsource the transportation function to carriers. Through an electronic auction a shipper can invite a selected set of carriers to submit a bid for a long-term contract to cover a given set of lanes. The auction can be convenient for both the shippers and the carriers. The shippers can identify the best offer, while the carriers can attract more and more shippers. Once long-term contracts are settled, the carriers often do not have an ideal situation, because often the trucks do not work with full load. Spot loads are an opportunity for carriers to fill up trucks or to avoid empty returns. A typical situation carriers face is that of having a fleet of trucks with pre-assigned routes, constructed on the basis of their regular customers. Through the web the carriers can find a number of spot loads they may decide whether to pick up or not. In some cases serving a customer may be convenient because it is compatible with the remaining capacity of a truck and implies a short de-tour or even no de-tour. In other cases, the additional cost for serving the customer is not compensated by the profit. The opportunity to pick up some of these potential customers has to be evaluated by looking at the full picture of the fleet of vehicles, of the planned and the potential customers. This problem can be modelled as a routing problem with profits where a fleet of capacitated vehicles is given as well as a set of customers that have to be served. In addition, a set of potential customers is available. The problem becomes to decide which of these potential customers to serve and how to construct the routes for the vehicles in such a way that a suitable objective function is optimized. In some cases the objective may be the maximization of the collected profit, given a limited time available for each vehicle. In some other cases the objective may be the maximization of the difference between the profit and the cost due to the distance travelled.

While the literature is rich in papers that study routing problems where all customers to be served are given (see Toth and Vigo, 2002), the number of papers on routing problems with profits is much more limited. The interest in these problems is only recently growing. The survey by Feillet *et al* (2005) is focused on the case where a single vehicle is available and defines those problems as traveling salesman problems (TSPs) with profits. The objective function may be the maximization

*Correspondence: *C Archetti, Department of Quantitative Methods, University of Brescia, C. da S. Chiara 50, Brescia 25122, Italy.*
E-mail: archetti@eco.unibs.it

of the collected total profit (orienteering problem, shortened to OP), the minimization of the total travelling cost (prize-collecting TSP) or the optimization of a combination of both (profitable tour problem, shortened to PTP). The few papers that treat the multiple vehicles case are cited in the survey. All such papers assume that the vehicles are un-capacitated. We call this class of problems the vehicle routing problems (VRPs) with profits.

In this paper, we investigate the capacitated versions of two known VRPs with profits. The first is the extension to the case of multiple tours of the OP. In the OP, given a set of potential customers with associated profit and given the distances between pairs of customers, the objective is to find the subset of customers for which the collected profit is maximum, given a constraint on the total length of the tour. The OP is also called the selective traveling salesman problem (STSP). The extension of the orienteering problem to the case of multiple tours is known as the team orienteering problem (TOP). In the TOP there is a time constraint on each tour. The TOP appeared in the literature in a paper by Butt and Cavalier (1994) under the name multiple tour maximum collection, problem, while the definition of TOP was introduced by Chao *et al* (1996). Two recent papers by Tang and Miller-Hooks (2005) and by Archetti *et al* (2007) proposed metaheuristics for the solution of the TOP. An exact algorithm for the TOP can be found in Boussier *et al* (2007). We study the capacitated version of the TOP (CTOP).

The second problem we study is the capacitated version of the PTP that we will refer to as capacitated PTP (CPTP). In the PTP (see Feillet *et al*, 2005) a single vehicle is available and the objective is to maximize the difference between the total collected profit and the cost of the total distance travelled. In the CPTP, the PTP is extended to the case where a fleet of capacitated vehicles is available.

Among the metaheuristics proposed for the solution of combinatorial optimization problems, the tabu search (see, eg, Gendreau *et al*, 1994) has been shown to be very effective for a variety of VRPs. Another interesting metaheuristic is the variable neighbourhood search (see Mladenović and Hansen, 1997). In this paper, the effectiveness of these metaheuristics is confirmed also on the VRPs with profits. We propose, both for the solution of the CTOP and of the CPTP, two variants of a tabu search algorithm and a variable neighbourhood search algorithm. These algorithms extend ideas presented in Archetti *et al* (2007) for the TOP. The heuristics are compared with exact methods based on column generation, which extend the general scheme described in Boussier *et al* (2007). While the exact algorithms can solve, for each of the two problems, instances of small size, the results show that the heuristics obtain very good results both for the CTOP and for the CPTP, in terms of solution quality, within a reasonable amount of time.

The paper is organized as follows. The CTOP and the CPTP are formalized in Section 2. The exact approach and the metaheuristics for each of the two problems are presented in Sections 3 and 4, respectively. The computational results are provided and commented in Section 5 and, finally, some conclusions are drawn.

2. The capacitated team orienteering problem and profitable tour problem

We consider a complete undirected graph $G = (V, E)$, where $V = \{1, \ldots, n\}$ is the set of vertices and E is the set of edges. Vertex 1 is a depot for m identical vehicles of capacity Q, while the remaining vertices represent potential customers. An edge $(i, j) \in E$ represents the possibility to travel from customer i to customer j. A non-negative demand d_i and a non-negative profit p_i is associated with each customer i ($d_1 = p_1 = 0$). A symmetric travel time t_{ij} and cost c_{ij} are associated with each edge $(i, j) \in E$. Each vehicle starts and ends its tour at vertex 1, and can visit any subset of customers with a total demand that does not exceed the capacity Q. The profit of each customer can be collected by one vehicle at most. In the following we suppose that $t_{ij} = c_{ij}$ for each edge (i, j).

In the capacitated team orienteering problem (CTOP) a subset of the potential customers available has to be selected. The objective is to maximize the total collected profit while satisfying, for each vehicle, a time limit $T_{\max}$ on the tour duration and the capacity constraint Q.

In the CPTP a subset of the potential customers available has to be selected with the objective of maximizing the difference between the total collected profit and the cost of the total distance travelled while satisfying the capacity constraint Q for each vehicle.

The case of regular customers that need to be visited can be included by modelling those customers as potential customers with a very large profit.

An example of a problem instance is provided in Figure 1a. Here $Q = 5$, $m = 2$, $d_i = 2$, $i = 2, \ldots, 6$, $c_{ij} = 2$ for each edge (i, j) except for edge $(2, 3)$ that has cost $c_{23} = 1$. For the CTOP the time limit $T_{\max}$ is equal to 5. In Figure 1b, the optimal solution of the CTOP is depicted while Figure 1c shows the optimal solution of the CPTP.

3. An exact approach to CTOP and CPTP

In this section, we present exact solution procedures for the CTOP and the CPTP. The two algorithms proposed, one for each problem, rely on a similar branch-and-price scheme. This scheme was first introduced for the TOP in Boussier *et al* (2007). The method is first adapted to the CTOP so that the additional capacity constraint can be handled. The adaptation to the CPTP is described subsequently. Some parts of the method are intentionally only sketched: the reader is referred to Boussier *et al* (2007) for further details. Practically, in both cases, the changes needed concern the subproblem (an elementary shortest path problem with resource constraints (ESPPRC)), where resource and cost definitions are adapted to the new structure.

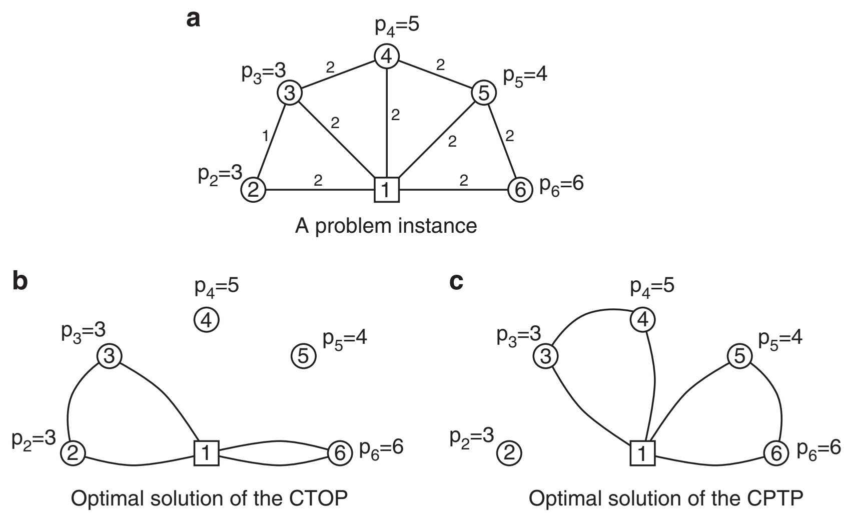

Figure 1 A problem instance and optimal solutions of the CTOP and CPTP.

3.1. Adaptation to the CTOP

Let $\Omega=\{r_1, \ldots, r_{|\Omega|}\}$ be the set of possible routes for a vehicle, that is, the set of routes starting and ending at vertex 1, visiting at most once each potential customer, satisfying the capacity constraint Q and the time limit $T_{\max}$. Let $a_{ik}=1$ if route $r_k \in \Omega$ visits customer i, $a_{ik}=0$ otherwise. Let c_k be the total profit generated by route $r_k \in \Omega$: $c_k=\sum_{i \in V} a_{ik} p_i$. The CTOP can be stated as follows:

$$\text{maximize} \quad \sum_{r_k \in \Omega} c_k x_k \tag{1}$$

subject to

$$\sum_{r_k \in \Omega} a_{ik} x_k \leqslant 1, \qquad i \in V \backslash \{1\} \tag{2}$$

$$\sum_{r_k \in \Omega} x_k \leqslant m \tag{3}$$

$$x_k \in \{0, 1\}, \qquad r_k \in \Omega \tag{4}$$

The decision variables x_k indicate whether route $r_k \in \Omega$ is used or not. Constraints (2) ensure that each customer is visited at most once. Constraint (3) limits the number of vehicles used to m. We denote by λ_i the non-negative dual variable associated with constraint (2) for customer i and by λ_1 the non-negative dual variable associated with constraint (3).

Solving the linear relaxation of model (1)–(4) necessitates the use of a column generation technique, due to the size of Ω. In the following, we call master problem (MP) the linear relaxation of model (1)–(4).

Column generation is based on two components: a restricted master problem and a subproblem. The restricted master problem $MP(\Omega_1)$ is obtained from MP by considering only a subset $\Omega_1 \subset \Omega$ of variables. The subproblem aims at adding progressively new potentially good columns to Ω_1 until an optimality criterion is attained. The reader is referred to Desaulniers *et al* (2005) for a recent book on the subject.

Ω_1 is initialized with a simple set of routes, each visiting a single customer (when possible). At each iteration of the algorithm, $MP(\Omega_1)$ is solved with the simplex method. The subproblem determines whether some variables x_k with $r_k \in \Omega \backslash \Omega_1$ have a positive reduced cost. This condition can easily be stated as

$$\sum_{i \in V \backslash \{1\}} a_{ik} \lambda_i + \lambda_1 < c_k$$

or equivalently,

$$\sum_{i \in r_k} (\lambda_i - p_i) < 0$$

where $i \in r_k$ means that vertex $i \in V$ is visited by route r_k. One or several variables with positive reduced cost are then added to Ω_1 and the algorithm iterates until the subproblem fails to find new routes.

The subproblem can be seen as an ESPPRC. To see that, one has to consider graph G and to introduce a cost $c'_{ij}=\lambda_i - p_i$ for each arc (i, j). Two resources l^1 and l^2 are then to be defined to take into account the capacity and the time constraints, respectively. Each arc (i, j) consumes a quantity $l^1_{ij}=d_i$ and $l^2_{ij}=t_{ij}$ of these resources; feasibility limits are set to Q and $T_{\max}$. The path has to be elementary in the sense that customers should not be visited more than once. The subproblem then consists in finding an elementary path, connecting 1 to 1, satisfying the two resource constraints and having a negative cost. Compared to the TOP, adding resource l^1 was needed here.

The ESPPRC can be solved through a dynamic programming approach, as proposed by Feillet *et al* (2004). One should note that the elementary path condition complicates the subproblem and could be removed. However, Ω would then be extended to paths visiting vertices several times, and therefore collecting profits several times, which would strongly weaken the quality of the upper bound provided by MP.

3.2. Adaptation to the CPTP

When addressing the CPTP instead of the CTOP, very few changes are needed. The set Ω is enlarged to include routes exceeding the time constraint $T_{\max}$. In the subproblem, the corresponding resource l^2 is removed. The cost c_k of a route r_k is changed to $c_k = \sum_{i \in r_k} p_i - \sum_{(i,j) \in r_k} c_{ij}$, where $(i, j) \in r_k$ means that arc (i, j) is used in r_k. The formula used to determine potentially good routes during the subproblem phase remains

$$\sum_{i \in V \setminus \{1\}} a_{ik}\lambda_i + \lambda_i < c_k$$

but can be rewritten as

$$\sum_{(i,j) \in r_k} (\lambda_i - p_i + c_{ij}) < 0$$

Hence, the definition of the arc costs when solving the ESPPRC is changed to $c'_{ij} = \lambda_i - p_i + c_{ij}$. No further transformation is needed.

3.3. Branching phase

In order to obtain the optimal solution of (1)–(4), column generation has to be embedded into a branching scheme. A usual branching rule, when using column generation for VRPs, is to select an arc (i, j) traversed by a fractional number of vehicles and to either enforce or forbid its traversal. In the case of VRPs with profits (as the CTOP and the CPTP), this rule raises some difficulties. The reason is that vertices i and j need not necessarily be visited. However, the branching rule can be adapted to handle this situation. When branching is needed, one can derive three branches: a first branch forbidding the visit of customer i, two other branches enforcing the visit of i followed or not by j. Also, this situation can be exploited. An additional branching rule can be defined as follows: when a customer i is visited a fractional number of times, a branch is created that enforces the visit of i, while a second branch forbids it. The branching first uses the latter rule when possible. Otherwise, the rule implying three branches is used. Further details on how these rules are implemented can be found in Boussier *et al* (2007). Note that neither the additional capacity constraint nor the particular objective of the CPTP is influential to this step of the algorithm.

3.4. Acceleration techniques

For the sake of efficiency, the branch-and-price algorithm described above is improved with a set of acceleration techniques. These acceleration techniques are deeply presented in Feillet *et al* (2005) and Boussier *et al* (2007). A first classical technique is to stop the ESPPRC resolution once a sufficiently large set of columns is found. A second acceleration technique is to start solving heuristically the ESPPRC, including only promising arcs with a Limited Discrepancy Search approach. A third technique is to take advantage of the property that a route $r_k \in \Omega_1$, such that $x_k > 0$ in the optimal solution of $MP(\Omega_1)$ has a reduced cost value equal to 0, which makes it a good starting point for finding new routes during the subproblem phase. This property is used in the so-called label loading and meta extend pre-processing procedures, which enhance the search towards these routes.

4. Metaheuristics for CTOP and CPTP

We have implemented three metaheuristics to solve the CTOP and the CPTP: a variable neighbourhood search algorithm and two tabu search algorithms. The tabu search algorithms differ from each other by the fact that one of them explores the feasible solution space, while the other enlarges this space to include also infeasible solutions. The main structure of the algorithms is the same as the one presented in Archetti *et al* (2007). In this work we adapted it to handle the capacity constraint and the different objectives of the CTOP and the CPTP.

The algorithms are first adapted for the CTOP so that the additional capacity constraint can be handled. The adaptation to the CPTP is described subsequently. We will only give a general description of the algorithms and focus mainly on the modifications we have made to the original algorithms in order to solve, respectively, the CTOP and the CPTP. A number of changes were needed to manage the solution space of the two problems. Among the major changes, we mention the way in which the infeasibility of a route is measured, the way in which the quality of a solution is measured and the procedures used to make, or try to make, an infeasible solution feasible. The reader is referred to (Archetti *et al*, 2007) for further details on the general structure.

4.1. Metaheuristics for CTOP

The profit $P(C)$ of a set $C \subseteq V$ of customers is the total profit $\sum_{i \in C} p_i$. The profit $P(r)$ of a route r is defined as the total profit of the customers on it, its duration $T(r)$ is its total travel time while $D(r)$ is the total demand of the customers visited by r. A route r is *feasible* if $T(r) \leqslant T_{\max}$ and $D(r) \leqslant Q$. To measure the possible infeasibility of a route r, we define $I_{\mathrm{CTOP}}(r) = \max\{D(r) - Q, 0\}^2 + \alpha * \max\{T(r) - T_{\max}, 0\}^2$, where $\alpha = \frac{T_{\max}}{Q}$. Hence, $I_{\mathrm{CTOP}}(r) = 0$ if and only if r is feasible. Parameter α is inserted in order to balance the two kinds of infeasibility. We define $I_{\mathrm{CTOP}}(r)$ as the sum of the squares of the two kinds of infeasibility since we prefer to have a solution containing different routes with a small infeasibility (and thus easy to correct) rather than a solution having few routes with a great infeasibility.

For a set R of routes, $P(R) = \sum_{r \in R} P(r)$ is defined as the total profit in R, $I_{\mathrm{CTOP}}(R) = \sum_{r \in R} I_{\mathrm{CTOP}}(r)$ is the total infeasibility in R and $C(R)$ is the set of customers visited on the routes in R.

A *solution* s is defined as a set of routes such that each route starts and ends at the depot, and each customer is visited exactly once by exactly one vehicle. We denote $R_{\mathrm{TOP}}(s)$ the set of m most profitable routes in s, and $R_{\mathrm{NTOP}}(s)$ the set of all remaining routes. A solution s is *feasible* if each route in s is feasible (ie $I_{\mathrm{CTOP}}(s) = 0$). A solution s is *admissible* if the routes in $R_{\mathrm{NTOP}}(s)$ are feasible (ie $I_{\mathrm{CTOP}}(R_{\mathrm{NTOP}}(s)) = 0$). Hence an admissible solution can have infeasible routes, but these are necessarily among the m most profitable ones. The aim of the CTOP is to determine a feasible solution s that maximizes $P(R_{\mathrm{TOP}}(s))$.

The customers in $C(R_{\mathrm{NTOP}}(s))$ do not belong to the most profitable routes in s, but are already organized into routes. It is therefore much easier to get a new route with a high profit by inserting new customers in one of the routes in $R_{\mathrm{NTOP}}(s)$ instead of creating a new route from scratch.

In a solution s, we denote $r_c(s)$ the route visiting customer c. For a customer c and a route $r \neq r_c(s)$, we denote $r + c$ the route obtained by adding c to r using the cheapest insertion technique. Similarly, given a route r and a customer c on r, we denote $r - c$ the route obtained from r by removing c and by linking its predecessor to its successor.

We have implemented two tabu search algorithms. One explores the set of feasible solutions while the other one visits admissible but not necessarily feasible solutions. They both use two kinds of moves:

- *1-move*: In a 1-move, customer c is moved from its route to a route $r \neq r_c(s)$. Route r can be an empty route. Hence, $r_c(s)$ and r are replaced by $r_c(s) - c$ and $r + c$, respectively.
- *swap-move*: Let c and c' be two customers on two different routes. A swap-move consists in replacing $r_c(s)$ and $r_{c'}(s)$ by $(r_c(s) - c) + c'$ and $(r_{c'}(s) - c') + c$, respectively.

A temporary tabu status forbids customers to be inserted into routes from which they have previously been removed. The solution s' resulting from a 1-move or a swap-move applied to s possibly contains infeasible routes. To remove the infeasibility in a subset R of routes, we use a repair procedure that performs a series of 1-moves that strictly reduce the infeasibility in R and do not modify the other existing routes. Notice that such 1-moves always exist since it is always possible to remove a customer from an infeasible route and to insert it into a new route. When exploring the set of admissible solutions, the infeasibility is removed only from the routes in $R_{\mathrm{NTOP}}(s')$, while all infeasible routes in s' must be repaired when exploring the set of feasible solutions.

The quality of feasible solutions is measured by combining the four following functions:

- $f_1(s)$: the total profit $P(R_{\mathrm{TOP}}(s))$ of the routes in $R_{\mathrm{TOP}}(s)$;
- $f_2(s)$: the total duration $\sum_{r \in R_{\mathrm{TOP}}(s)} T(r)$ of the routes in $R_{\mathrm{TOP}}(s)$;
- $f_3(s)$: the number of non-empty routes in s;
- $f_4(s)$: the total duration $\sum_{r \in R_{\mathrm{NTOP}}(s)} T(r)$ of the routes in $R_{\mathrm{NTOP}}(s)$.

A feasible solution s is considered as better than a feasible solution s' if $f_1(s) > f_1(s')$, or $f_1(s) = f_1(s')$ and $f_2(s) < f_2(s')$, or $f_1(s) = f_1(s')$, $f_2(s) = f_2(s')$ and $f_3(s) < f_3(s')$, or $f_1(s) = f_1(s')$, $f_2(s) = f_2(s')$, $f_3(s) = f_3(s')$ and $f_4(s) < f_4(s')$.

When exploring the set of admissible solutions, a fifth function $f_5(s)$ replaces functions $f_1(s)$ and $f_2(s)$. It is defined as $f_5(s) = P(R_{\mathrm{TOP}}(s)) - \beta * \max\{T(r) - T_{\max}, 0\}^2 - \gamma * \max\{D(r) - Q, 0\}^2$, where β and γ are parameters that give more or less importance to the capacity and duration infeasibility, respectively. Notice that $f_5(s) = f_1(s)$ if s is feasible. Parameters β and γ are initially set equal to 1 and are then adjusted every 10 iterations, as in Gendreau *et al* (1994): if the 10 previous solutions were feasible with respect to the capacity (duration), then $\gamma(\beta)$ is divided by 2; if they were all infeasible, then $\gamma(\beta)$ is multiplied by 2; otherwise, $\gamma(\beta)$ remains unchanged. An admissible solution s is considered as better than an admissible solution s' if $f_5(s) > f_5(s')$, or $f_5(s) = f_5(s')$ and $f_3(s) < f_3(s')$, or $f_5(s) = f_5(s')$, $f_3(s) = f_3(s')$ and $f_4(s) < f_4(s')$.

In order to explore various regions of the solution space, we use two kinds of jumps to escape a local optimum. One consists in performing a series of 1-moves from $R_{\mathrm{NTOP}}(s)$ to $R_{\mathrm{TOP}}(s)$, while the other moves a set U of customers from $R_{\mathrm{TOP}}(s)$ to $R_{\mathrm{NTOP}}(s)$, and a set W of customers from $R_{\mathrm{NTOP}}(s)$ to $R_{\mathrm{TOP}}(s)$, so that $P(U) \leqslant P(W)$. Again, a solution s' resulting from a jump possibly contains infeasible routes. When exploring the set of admissible solutions, the infeasibility is removed only from the routes in $R_{\mathrm{NTOP}}(s')$, while all infeasible routes in s' must be repaired when exploring the set of feasible solutions. For this purpose, we again use the repair procedure described above.

The two tabu search algorithms typically visit a limited number of different regions of the solution space, each one being deeply explored before jumping to the next one. On the contrary, the variable neighbourhood search algorithm performs many more jumps, which results in an exploration of a larger number of different regions of the solution space, but at the expense of a less intensive search in each region. The variable neighbourhood search algorithm explores the feasible solution space, using the above tabu search algorithm for a very limited number of iterations in each region.

4.2. *Metaheuristics for CPTP*

The value $V(r)$ of a route r is defined as the total profit of the customers on it minus its duration, that is, $V(r) = P(r) - T(r)$. A route r is *feasible* if $D(r) \leqslant Q$. To measure the possible infeasibility of a route r, we define $I_{\mathrm{CPTP}}(r) = \max\{D(r) - Q, 0\}^2$. For a set R of routes, $V(R) = \sum_{r \in R} V(r)$ denotes the total value of the routes in R. We denote $R_{\mathrm{PTP}}(s)$ the set of m routes with the largest value in s, and $R_{\mathrm{NPTP}}(s)$ the set of all remaining routes. The aim of the CPTP is to determine a feasible solution s that maximizes $V(R_{\mathrm{PTP}}(s))$.

The two tabu search algorithms and the variable neighbourhood search algorithm we have implemented for the CPTP are identical to those developed for the CTOP, except that $f_1(s)$ is defined as the total value $V(R_{\text{PTP}}(s))$ of the routes in $R_{\text{PTP}}(s)$ and $f_5(s)$ is defined as $V(R_{\text{PTP}}(s)) - \delta * I_{\text{CPTP}}(R_{\text{PTP}}(s))$, where δ is a parameter that gives more or less importance to the second component of this function.

Note that $R_{\text{PTP}}(s)$ contains only routes r with a strictly positive value $V(r)$, even when the number of these routes in the solution s is smaller than m.

5. Computational results

In what follows, we call *VNS* the variable neighbourhood search, *TabuFeasible* the tabu search that explores the set of feasible solutions and *TabuAdmissible* the tabu search that explores the set of admissible solutions. The branch-and-price algorithm was tested on a Intel Pentium 4 CPU 1.60 GHz and 256 MB Ram. The code is written in C++ and the exact solver used is Cplex 9.0. The three heuristics are coded in Visual C++ and run on a personal computer Intel Pentium 4 CPU 2.80 GHz and 1.048 GB Ram. All the algorithms were stopped after one hour of computational time.

To evaluate the algorithms we used 10 Christofides *et al*'s (1979) instances taken from the VRP library that report both capacity and time constraints. The number n of vertices ranges from 51 to 200. We have defined the profit p_i of customer i as equal to $(0.5 + h)d_i$, where h is a random number uniformly generated in the interval [0, 1]. Starting from these 10 basic instances we have considered three sets of tests.

The first set of tests contains the original instances, that is, without modification of the capacity and time limits and the number of available vehicles. The results are shown in Table 1 for the CTOP and in Table 2 for the CPTP. The '−' on columns CPU means that the time limit of one hour has been reached. For each instance, we indicate the number n of vertices, the number m of vehicles, the capacity limit Q and the time limit $T_{\max}$ (for the CTOP). The optimal profit of the CTOP determined by the branch-and-price algorithm is indicated under column p^* of B&P in Table 1. A value in italic means that the optimal value could not be obtained within a time limit of one hour. In such a case, the value represents the best lower bound generated by the exact method during the search process. Column CPU of B&P reports the computational time (in seconds) of the branch-and-price algorithm. The results for the three heuristics appear under columns VNS (for the *VNS* algorithm), Tabu Feasible (for the *TabuFeasible* algorithm) and Tabu Admissible (for the *TabuAdmissible* algorithm). For each heuristic, we indicate the best found profit (column p), the computational time in seconds (column CPU) and the percentage deviation from the best known solution value (column %), which is defined as the maximum value appearing under columns labelled p^* and p. A bold value under a column p means that the heuristic has found the best known solution. For the CPTP (Table 2), we do not report the results of the exact method since it has not found any optimal solution within one hour of computation. The other information has the same meaning as in Table 1, except that the profit p is replaced by value v corresponding to the total profit minus the total duration.

These 10 original instances have a quite large number m of vehicles (since they are benchmark instances for the VRP). As a consequence, the CTOP is almost similar to a multiple knapsack problem. We observe from Table 1 that the three heuristics determine their best solutions in less than 1 second for eight of the 10 instances. This is due to the fact that all customers can be visited with the m vehicles without exceeding the capacity and time limits. As a consequence, the optimal solution value of the CTOP is the total profit $\sum_{i \in V \setminus \{1\}} p_i$. Since this is a trivial upper bound on the optimal value of the CTOP, we stop the heuristic algorithms when it is reached. The CPTP appears to be much more difficult to solve on these instances, the reason being that in addition to visiting all customers, this should be done in an optimal way, and the number of feasible solutions grows exponentially with the number of vehicles.

The three heuristics produce comparable results for the CTOP, while for the CPTP, the best performance is achieved with the *VNS* algorithm that finds all best known solutions, except for instances 9 and 16. *VNS* is, however, much slower than the tabu search algorithms that produce solutions of reasonably good quality in a much shorter time. There is a slight advantage to explore the set of feasible solutions rather than the set of admissible solutions.

In order to better compare the performance of the heuristic algorithms with the optimal solutions produced by the branch-and-price algorithm, we changed the data referring to the capacity and time limits. In particular, for the CPTP, from each basic instance we consider the cases with $Q = 50$, 75 and 100, respectively, while for the CTOP we consider the cases with $Q = 50$ and $T_{\max} = 50$, $Q = 75$ and $T_{\max} = 75$ and $Q = 100$ and $T_{\max} = 100$, respectively. Moreover, for each case, we generated three different instances with $m = 2$, 3, 4, respectively, for a total of 90 instances for each problem. The results of this second set of tests are reported in Table 3 for the CTOP and Table 4 for the CPTP. The meaning of the columns is identical as in Table 1 and 2.

We observe from Table 3 that the branch-and-price algorithm has determined the optimal solution of 60 out of the 90 instances of the CTOP. *TabuFeasible* is the best of the three heuristics. It has produced 79 best known solutions, including 57 proved optima. For comparison, *VNS* has produced 72 best known solutions with 53 proved optima, while *TabuAdmissible* has found 54 best known solutions with 45 proved optima. Notice, however, that the three heuristics produce results of similar quality since the average percentage deviation from the best known solution values ranges from 0.06 to 0.21. The computing times of the two tabu search heuristics are comparable, with a slight advantage for the exploration of the set of feasible solutions, while *VNS* is a little bit slower.

Table 4 clearly shows that *VNS* is the best among the three heuristics for the CPTP. It finds the best known solution for all 90 instances. For comparison, *TabuFeasible* finds 65 best

Table 1 Results for the CTOP on the original test set

Instance					***B&P***		***VNS***			***Tabu Feasible***			***Tabu Admissible***		
Instance	*n*	*m*	*Q*	$T_{\max}$	p^*	*CPU*	*p*	*CPU*	*%*	*p*	*CPU*	*%*	*p*	*CPU*	*%*
3	101	15	200	200	1409	41	**1409**	0	0.00	**1409**	0	0.00	**1409**	0	0.00
6	51	10	160	200	761	2	**761**	0	0.00	**761**	0	0.00	**761**	0	0.00
7	76	20	140	160	1327	2	**1327**	0	0.00	**1327**	0	0.00	**1327**	0	0.00
8	101	15	200	230	1409	17	**1409**	0	0.00	**1409**	0	0.00	**1409**	0	0.00
9	151	10	200	200	*1164*	–	**2064**	–	0.00	2061	163	0.15	2062	127	0.10
10	200	20	200	200	*1735*	–	**3048**	0	0.00	**3048**	0	0.00	**3048**	0	0.00
13	121	15	200	720	1287	21	**1287**	0	0.00	**1287**	0	0.00	**1287**	0	0.00
14	101	10	200	1040	1710	1082	**1710**	0	0.00	**1710**	0	0.00	**1710**	0	0.00
15	151	15	200	200	2159	1866	**2159**	0	0.00	**2159**	0	0.00	**2159**	0	0.00
16	200	15	200	200	*588*	–	**2968**	–	0.00	2965	270	0.10	2967	377	0.03
Average									0.00			0.02			0.01

Table 2 Results for the CPTP on the original test set

Instance				***VNS***			***Tabu Feasible***			***Tabu Admissible***		
Instance	*n*	*m*	*Q*	*v*	*CPU*	*%*	*v*	*CPU*	*%*	*v*	*CPU*	*%*
3	101	15	200	**663.98**	400	0.00	657.31	39	1.01	656.32	13	1.17
6	51	10	160	**258.97**	22	0.00	**258.97**	7	0.00	255.38	2	1.41
7	76	20	140	**534.81**	198	0.00	525.06	23	1.86	527.90	13	1.31
8	101	15	200	**663.98**	400	0.00	657.31	40	1.01	656.32	13	1.17
9	151	10	200	1189.33	–	0.28	**1192.68**	162	0.00	1143.65	45	4.29
10	200	20	200	**1773.65**	1384	0.00	1761.37	420	0.70	1759.81	110	0.79
13	121	15	200	**284.71**	219	0.00	269.74	52	5.55	274.28	66	3.80
14	101	10	200	**890.44**	145	0.00	886.78	51	0.41	888.18	19	0.25
15	151	15	200	**1168.63**	3298	0.00	1156.01	199	1.09	1134.17	42	3.04
16	200	15	200	1776.09	–	0.02	1764.15	365	1.57	**1776.41**	139	0.00
Average						0.03			1.23			1.72

Table 3 Results for the CTOP on the second test set

Instance					***EXACT***		***VNS***			***Tabu Feasible***			***Tabu Admissible***		
Instance	*n*	*m*	*Q*	T_{max}	p^*	*CPU*	*p*	*CPU*	*%*	*p*	*CPU*	*%*	*p*	*CPU*	*%*
3	101	2	50	50	133	1	**133**	46	0.00	**133**	34	0.00	**133**	49	0.00
3	101	3	50	50	198	0	**198**	63	0.00	**198**	34	0.00	**198**	85	0.00
3	101	4	50	50	260	1	**260**	89	0.00	**260**	35	0.00	**260**	39	0.00
3	101	2	75	75	208	65	**208**	383	0.00	**208**	224	0.00	**208**	290	0.00
3	101	3	75	75	307	325	**307**	500	0.00	**307**	225	0.00	**307**	304	0.00
3	101	4	75	75	403	88	401	685	0.50	**403**	299	0.00	402	442	0.25
3	101	2	100	100	*277*	–	**277**	472	0.00	**277**	291	0.00	276	304	0.36
3	101	3	100	100	*407*	–	407	685	0.25	**408**	320	0.00	407	356	0.25
3	101	4	100	100	*526*	–	529	963	0.38	**531**	317	0.00	529	357	0.38
6	51	2	50	50	121	0	**121**	5	0.00	**121**	3	0.00	**121**	4	0.00
6	51	3	50	50	177	0	**177**	8	0.00	**177**	3	0.00	**177**	5	0.00
6	51	4	50	50	222	0	**222**	13	0.00	**222**	3	0.00	**222**	5	0.00
6	51	2	75	75	183	1	**183**	53	0.00	**183**	33	0.00	**183**	31	0.00
6	51	3	75	75	269	0	**269**	71	0.00	**269**	28	0.00	**269**	27	0.00
6	51	4	75	75	349	1	**349**	96	0.00	348	25	0.29	348	24	0.29
6	51	2	100	100	252	283	**252**	60	0.00	**252**	28	0.00	251	34	0.40
6	51	3	100	100	369	2696	**369**	92	0.00	**369**	30	0.00	**369**	28	0.00
6	51	4	100	100	482	212	481	135	0.21	**482**	25	0.00	481	26	0.21
7	76	2	50	50	126	0	**126**	17	0.00	**126**	14	0.00	**126**	22	0.00
7	76	3	50	50	187	0	**187**	27	0.00	**187**	14	0.00	**187**	19	0.00
7	76	4	50	50	240	1	**240**	40	0.00	**240**	13	0.00	**240**	14	0.00
7	76	2	75	75	193	1	**193**	149	0.00	**193**	89	0.00	**193**	127	0.00

Table 3 Continued

Instance					***EXACT***		***VNS***			***Tabu Feasible***			***Tabu Admissible***		
Instance	*n*	*m*	*Q*	T_{max}	p^*	*CPU*	*p*	*CPU*	*%*	*p*	*CPU*	*%*	*p*	*CPU*	*%*
7	76	3	75	75	287	1	**287**	183	0.00	**287**	87	0.00	**287**	164	0.00
7	76	4	75	75	378	2	**378**	265	0.00	**378**	88	0.00	377	111	0.27
7	76	2	100	100	266	244	**266**	163	0.00	**266**	95	0.00	**266**	137	0.00
7	76	3	100	100	397	367	**397**	233	0.00	**397**	113	0.00	391	137	1.53
7	76	4	100	100	521	1733	**521**	342	0.00	**521**	119	0.00	514	148	1.36
8	101	2	50	50	133	1	**133**	46	0.00	**133**	34	0.00	**133**	49	0.00
8	101	3	50	50	198	0	**198**	59	0.00	**198**	34	0.00	**198**	85	0.00
8	101	4	50	50	260	0	**260**	88	0.00	**260**	35	0.00	**260**	39	0.00
8	101	2	75	75	208	62	**208**	383	0.00	**208**	224	0.00	**208**	290	0.00
8	101	3	75	75	307	321	**307**	500	0.00	**307**	225	0.00	**307**	304	0.00
8	101	4	75	75	403	88	401	685	0.50	403	299	0.00	402	442	0.25
8	101	2	100	100	*277*	–	**277**	472	0.00	**277**	291	0.00	276	304	0.36
8	101	3	100	100	*407*	–	407	685	0.25	**408**	320	0.00	407	356	0.25
8	101	4	100	100	*526*	–	529	963	0.38	**531**	317	0.00	529	357	0.38
9	151	2	50	50	137	1	**137**	192	0.00	**137**	115	0.00	**137**	166	0.00
9	151	3	50	50	201	0	**201**	227	0.00	**201**	115	0.00	**201**	284	0.00
9	151	4	50	50	262	3	**262**	293	0.00	**262**	112	0.00	**262**	178	0.00
9	151	2	75	75	210	1178	**210**	1166	0.00	**210**	785	0.00	**210**	1411	0.00
9	151	3	75	75	312	1166	310	1724	0.65	310	808	0.65	310	1581	0.65
9	151	4	75	75	408	1266	407	1862	0.25	407	958	0.25	407	1026	0.25
9	151	2	100	100	*279*	–	**279**	1646	0.00	278	971	0.36	**279**	1099	0.00
9	151	3	100	100	*413*	–	413	2151	0.24	**414**	1521	0.00	412	1147	0.49
9	151	4	100	100	*543*	–	**545**	2934	0.00	539	924	1.11	536	959	1.68
10	200	2	50	50	134	8	**134**	423	0.00	**134**	312	0.00	**134**	477	0.00
10	200	3	50	50	200	7	200	549	0.00	**200**	303	0.00	**200**	535	0.00
10	200	4	50	50	265	3	**265**	788	0.00	**265**	324	0.00	264	628	0.38
10	200	2	75	75	208	1860	**208**	2596	0.00	**208**	1759	0.00	**208**	–	0.00
10	200	3	75	75	*311*	–	310	3122	0.32	310	1890	0.32	310	2742	0.32
10	200	4	75	75	*410*	–	**410**	–	0.00	**410**	2194	0.00	407	–	0.74
10	200	2	100	100	*281*	–	**282**	3111	0.00	280	2115	0.71	279	–	1.08
10	200	3	100	100	*407*	–	416	–	0.24	**417**	2606	0.00	412	3122	1.21
10	200	4	100	100	*552*	–	548	–	0.73	549	2077	0.55	550	3232	0.36
13	121	2	50	50	134	18	**134**	21	0.00	**134**	10	0.00	**134**	10	0.00
13	121	3	50	50	193	96	**193**	30	0.00	**193**	9	0.00	**193**	11	0.00
13	121	4	50	50	243	177	**243**	47	0.00	**243**	8	0.00	242	10	0.41
13	121	2	75	75	*185*	–	**193**	11	0.00	**193**	4	0.00	**193**	3	0.00
13	121	3	75	75	*255*	–	**265**	18	0.00	**265**	4	0.00	**265**	3	0.00
13	121	4	75	75	*323*	–	**323**	23	0.00	**323**	4	0.00	**323**	3	0.00
13	121	2	100	100	*251*	–	**253**	76	0.00	**253**	27	0.00	**253**	35	0.00
13	121	3	100	100	*321*	–	**344**	113	0.00	**344**	24	0.00	343	31	0.29
13	121	4	100	100	*340*	–	**419**	179	0.00	**419**	24	0.00	416	48	0.72
14	101	2	50	50	124	0	**124**	46	0.00	**124**	37	0.00	**124**	40	0.00
14	101	3	50	50	184	0	**184**	67	0.00	**184**	35	0.00	**184**	52	0.00
14	101	4	50	50	241	0	**241**	97	0.00	**241**	36	0.00	**241**	49	0.00
14	101	2	75	75	190	24	190	140	0.00	**190**	98	0.00	190	146	0.00
14	101	3	75	75	279	34	**279**	201	0.00	**279**	97	0.00	**279**	169	0.00
14	101	4	75	75	366	64	**366**	316	0.00	**366**	102	0.00	366	194	0.00
14	101	2	100	100	*271*	–	**271**	276	0.00	**271**	181	0.00	**271**	423	0.00
14	101	3	100	100	*377*	–	**399**	406	0.00	**399**	248	0.00	399	302	0.00
14	101	4	100	100	*525*	–	**525**	670	0.00	523	210	0.38	525	292	0.00
15	151	2	50	50	134	0	**134**	166	0.00	**134**	123	0.00	133	163	0.75
15	151	3	50	50	200	0	**200**	214	0.00	**200**	129	0.00	199	157	0.50
15	151	4	50	50	266	0	**266**	305	0.00	**266**	113	0.00	**266**	202	0.00
15	151	2	75	75	211	736	210	1240	0.48	**211**	904	0.00	**211**	1277	0.00
15	151	3	75	75	315	743	**315**	1593	0.00	**315**	782	0.00	**315**	934	0.00
15	151	4	75	75	*412*	–	**414**	2118	0.00	**414**	998	0.00	413	1167	0.24
15	151	2	100	100	*282*	–	**282**	1672	0.00	**282**	924	0.00	**282**	1316	0.00
15	151	3	100	100	*410*	–	**417**	1966	0.00	416	883	0.24	416	1248	0.24
15	151	4	100	100	*546*	–	548	2828	0.18	**549**	1252	0.00	545	1015	0.73
16	200	2	50	50	137	1	**137**	422	0.00	**137**	305	0.00	**137**	529	0.00

Table 3 Continued

Instance					***EXACT***		***VNS***			***Tabu Feasible***			***Tabu Admissible***		
Instance	n	m	Q	T_{max}	p^*	*CPU*	p	*CPU*	%	p	*CPU*	%	p	*CPU*	%
16	200	3	50	50	203	1	**203**	580	0.00	**203**	303	0.00	**203**	455	0.00
16	200	4	50	50	269	1	**269**	677	0.00	**269**	295	0.00	**269**	472	0.00
16	200	2	75	75	212	1860	**212**	2808	0.00	**212**	1821	0.00	**212**	2935	0.00
16	200	3	75	75	317	1857	**317**	–	0.00	**317**	2156	0.00	**317**	–	0.00
16	200	4	75	75	420	2525	419	–	0.24	**420**	2435	0.00	419	–	0.24
16	200	2	100	100	*285*	–	284	3523	0.35	**285**	2144	0.00	284	3234	0.35
16	200	3	100	100	*422*	–	420	–	0.48	421	2421	0.24	421	–	0.24
16	200	4	100	100	*544*	–	**554**	–	0.00	**554**	2124	0.00	553	3559	0.18
Average									0.07			0.06			0.21

Table 4 Results for the CPTP on the second test set

Instance				***EXACT***		***VNS***			***Tabu Feasible***			***Tabu Admissible***		
Instance	n	m	Q	p^*	*CPU*	p	*CPU*	%	p	*CPU*	%	p	*CPU*	%
3	101	2	50	57.75	6	**57.75**	84	0.00	**57.75**	50	0.00	**57.75**	165	0.00
3	101	3	50	80.82	6	**80.82**	108	0.00	**80.82**	51	0.00	**80.82**	200	0.00
3	101	4	50	100.36	7	**100.36**	97	0.00	98.47	49	1.93	**100.36**	200	0.00
3	101	2	75	106.15	519	**106.15**	106	0.00	**106.15**	38	0.00	**106.15**	160	0.00
3	101	3	75	147.55	450	**147.55**	114	0.00	**147.55**	46	0.00	145.87	255	1.15
3	101	4	75	185.27	248	**185.27**	119	0.00	**185.27**	45	0.00	**185.27**	173	0.00
3	101	2	100	*158.21*	–	**158.21**	117	0.00	**158.21**	43	0.00	**158.21**	149	0.00
3	101	3	100	**218.63**	–	**218.63**	134	0.00	**218.63**	37	0.00	*218.33*	102	0.14
3	101	4	100	*255.33*	–	**268.34**	156	0.00	266.23	38	0.79	266.08	106	0.85
6	51	2	50	33.88	0	**33.88**	9	0.00	**33.88**	6	0.00	**33.88**	22	0.00
6	51	3	50	40.95	1	**40.95**	11	0.00	**40.95**	7	0.00	**40.95**	27	0.00
6	51	4	50	45.43	0	**45.43**	10	0.00	**45.43**	6	0.00	**45.43**	35	0.00
6	51	2	75	72.28	4	**72.28**	10	0.00	**72.28**	6	0.00	**72.28**	11	0.00
6	51	3	75	92.32	3	**92.32**	11	0.00	**92.32**	6	0.00	**92.32**	16	0.00
6	51	4	75	99.37	57	**99.37**	12	0.00	**99.37**	6	0.00	**99.37**	12	0.00
6	51	2	100	100.27	159	**100.27**	14	0.00	99.50	4	0.77	99.50	11	0.77
6	51	3	100	134.72	119	**134.72**	13	0.00	**134.72**	5	0.00	**134.72**	10	0.00
6	512	4	100	153.30	1774	**153.30**	16	0.00	**153.30**	8	0.00	152.97	7	0.22
7	76	2	50	49.18	1	**49.18**	39	0.00	**49.18**	28	0.00	**49.18**	100	0.00
7	76	3	50	69.94	0	**69.94**	42	0.00	**69.94**	26	0.00	**69.94**	90	0.00
7	76	4	50	90.65	1	**90.65**	42	0.00	**90.65**	26	0.00	**90.65**	121	0.00
7	76	2	75	92.44	5	**92.44**	45	0.00	**92.44**	23	0.00	**92.44**	104	0.00
7	76	3	75	131.12	9	**131.12**	64	0.00	**131.12**	25	0.00	**131.12**	80	0.00
7	76	4	75	*158.11*	39	**158.11**	65	0.00	**158.11**	25	0.00	**158.11**	94	0.00
7	76	2	100	132.70	441	**132.70**	53	0.00	**132.70**	24	0.00	**132.70**	54	0.00
7	76	3	100	185.25	710	**185.25**	72	0.00	184.88	20	0.20	**185.25**	83	0.00
7	76	4	100	233.40	684	**233.40**	77	0.00	**233.40**	29	0.00	**232.05**	79	0.58
8	101	2	50	57.75	6	**57.75**	85	0.00	**57.75**	50	0.00	**57.75**	165	0.00
8	101	3	50	80.82	6	**80.82**	108	0.00	**80.82**	51	0.00	**80.82**	200	0.00
8	101	4	50	100.36	7	**100.36**	97	0.00	98.47	49	1.93	**100.36**	200	0.00
8	101	2	75	106.15	511	**106.15**	106	0.00	**106.15**	38	0.00	**106.15**	160	0.00
8	101	3	75	147.55	450	**147.55**	114	0.00	**147.55**	46	0.00	145.87	255	1.15
8	101	4	75	185.27	248	**185.27**	119	0.00	**185.27**	45	0.00	**185.27**	173	0.00
8	101	2	100	*158.21*	–	**158.21**	117	0.00	**158.21**	43	0.00	**158.21**	148	0.00
8	101	3	100	*218.63*	–	**218.63**	134	0.00	**218.63**	37	0.00	218.33	102	0.14
8	101	4	100	*255.33*	–	**268.34**	157	0.00	266.23	39	0.79	266.08	106	0.85
9	151	2	50	65.03	34	**65.03**	369	0.00	63.89	211	1.78	**65.03**	966	0.00
9	151	3	50	96.16	34	**96.16**	489	0.00	**96.16**	206	0.00	**96.16**	826	0.00
9	151	4	50	121.35	35	**121.35**	562	0.00	**121.35**	197	0.00	**121.35**	783	0.00
9	151	2	75	117.66	2666	**117.66**	447	0.00	**117.66**	171	0.00	**117.66**	618	0.00
9	151	3	75	*160.66*	–	**160.96**	566	0.00	**160.96**	197	0.00	**160.96**	634	0.00
9	151	4	75	204.25	2862	**204.25**	735	0.00	203.24	238	0.50	203.24	639	0.50
9	151	2	100	*161.15*	–	**161.23**	462	0.00	**161.23**	168	0.00	**161.23**	456	0.00

Table 4 Continued

Instance				***EXACT***		***VNS***			***Tabu Feasible***			***Tabu Admissible***		
Instance	*n*	*m*	*Q*	*p**	*CPU*	*p*	*CPU*	*%*	*p*	*CPU*	*%*	*p*	*CPU*	*%*
9	151	3	100	*226.29*	–	**230.49**	615	0.00	229.61	197	0.38	229.61	801	0.38
9	151	4	100	*286.39*	–	**290.54**	876	0.00	**290.54**	294	0.00	290.15	729	0.13
10	200	2	50	70.87	98	**70.87**	1009	0.00	**70.87**	628	0.00	**70.87**	2632	0.00
10	200	3	50	103.79	85	**103.79**	1332	0.00	**103.79**	633	0.00	**103.79**	2260	0.00
10	200	4	50	134.81	85	**134.81**	1563	0.00	**134.81**	659	0.00	**134.81**	–	0.00
10	200	2	75	*124.85*	–	**124.85**	1219	0.00	**124.85**	532	0.00	**124.85**	1822	0.00
10	200	3	75	*177.90*	–	177.90	1526	0.00	**177.90**	517	0.00	176.50	2248	0.79
10	200	4	75	*229.27*	–	**229.27**	2008	0.00	**229.27**	528	0.00	**229.27**	1606	0.00
10	200	2	100	*171.20*	–	**171.24**	1306	0.00	**171.24**	512	0.00	**171.24**	1962	0.00
10	200	3	100	*246.56*	–	**250.18**	1748	0.00	246.56	486	1.47	246.95	1858	1.31
10	200	4	100	*322.32*	–	**324.02**	2299	0.00	321.17	515	0.89	321.03	1433	0.93
13	121	2	50	*63.92*	–	**64.12**	183	0.00	**64.12**	97	0.00	**64.12**	410	0.00
13	121	3	50	*85.72*	–	**87.25**	188	0.00	**87.25**	90	0.00	**87.25**	419	0.00
13	121	4	50	*94.84*	–	**104.18**	219	0.00	103.73	126	0.43	103.72	363	0.44
13	121	2	75	*110.12*	–	**110.12**	198	0.00	**110.12**	77	0.00	**110.12**	122	0.00
13	121	3	75	*135.10*	–	**139.37**	192	0.00	137.95	71	1.03	137.45	218	1.40
13	121	4	75	*156.64*	–	**161.62**	224	0.00	160.68	78	0.59	157.98	120	2.31
13	121	2	100	*145.75*	–	**145.75**	186	0.00	145.67	78	0.06	145.67	138	0.06
13	121	3	100	*180.79*	–	**181.63**	162	0.00	177.76	74	2.18	180.04	112	0.88
13	121	4	100	*197.86*	–	**200.62**	158	0.00	178.82	77	12.19	183.66	145	9.23
14	101	2	50	43.26	5	**43.26**	116	0.00	**43.26**	60	0.00	**43.26**	267	0.00
14	101	3	50	59.43	7	**59.43**	143	0.00	**59.43**	67	0.00	**59.43**	210	0.00
14	101	4	50	68.63	10	**68.63**	150	0.00	**68.63**	68	0.00	**68.63**	206	0.00
14	101	2	75	77.09	78	**77.09**	115	0.00	**77.09**	59	0.00	**77.09**	247	0.00
14	101	3	75	112.56	79	**112.56**	150	0.00	112.51	61	0.04	**112.56**	218	0.00
14	101	4	75	139.88	138	**139.88**	171	0.00	139.67	71	0.15	139.83	241	0.03
14	101	2	100	125.29	–	**125.29**	121	0.00	**125.29**	51	0.00	**125.29**	164	0.00
14	101	3	100	*182.31*	–	**182.31**	185	0.00	179.48	63	1.58	**182.31**	211	0.00
14	101	4	100	*236.29*	–	**237.68**	224	0.00	236.50	57	0.50	**237.68**	198	0.00
15	151	2	50	64.98	39	**64.98**	359	0.00	**64.98**	202	0.00	**64.98**	1210	0.00
15	151	3	50	96.42	41	**96.42**	460	0.00	**96.42**	205	0.00	**96.42**	844	0.00
15	151	4	50	124.02	41	**124.02**	569	0.00	**124.02**	195	0.00	**124.02**	773	0.00
15	151	2	75	120.93	2536	**120.93**	412	0.00	**120.93**	170	0.00	**120.93**	662	0.00
15	151	3	75	174.58	2705	**174.58**	557	0.00	**174.58**	183	0.00	**174.58**	706	0.00
15	151	4	75	219.22	2626	**219.22**	669	0.00	219.22	173	0.00	216.61	645	1.21
15	151	2	100	*166.93*	–	**169.71**	438	0.00	**169.71**	175	0.00	**169.71**	675	0.00
15	151	3	100	*238.39*	–	**244.08**	597	0.00	241.84	172	0.93	**244.08**	573	0.00
15	151	4	100	*306.39*	–	**308.07**	896	0.00	305.30	168	0.91	304.81	508	1.07
16	200	2	50	66.81	90	**66.81**	1009	0.00	**66.81**	645	0.00	**66.81**	2444	0.00
16	200	3	50	99.70	90	**99.70**	1278	0.00	**99.70**	650	0.00	**99.70**	2687	0.00
16	200	4	50	131.37	91	**131.37**	1532	0.00	**131.37**	640	0.00	**131.37**	2994	0.00
16	200	2	75	*123.38*	–	**123.38**	1255	0.00	**123.38**	539	0.00	**123.38**	1964	0.00
16	200	3	75	*179.55*	–	**179.55**	1597	0.00	**179.55**	548	0.00	179.23	1728	0.18
16	200	4	75	*235.03*	–	**235.03**	2034	0.00	**235.03**	577	0.00	**235.03**	1580	0.00
16	200	2	100	*173.56*	–	**177.23**	1216	0.00	**177.23**	732	0.00	175.57	1662	0.94
16	200	3	100	*239.34*	–	**258.07**	1590	0.00	257.10	610	0.38	252.44	1694	2.23
16	200	4	100	*330.14*	–	**336.24**	2214	0.00	328.20	488	2.45	329.53	1191	2.04
Average								0.00			0.39			0.35

known solutions, 45 being proved optima, while *TabuAdmissible* produces 62 best known solutions with also 45 proved optima. The average percentage deviation from the best known solution values is less than 0.4 for both tabu search algorithms. *VNS* is again the slowest algorithm and *TabuFeasible* the fastest. For these instances, the branch-and-price algorithm has produced 53 optimal solutions within the one hour limit.

A third set of tests is made on the original data, changing only the number of vehicles. For each instance we consider the cases with $m = 2, 3, 4$, respectively, for a total of 30 instances for each problem. These tests are made in order to avoid the previous situation where the CTOP becomes similar to a multiple knapsack problem while maintaining the original data, and thus the original complexity of the instances. For these tests we simply compare the results of the heuristic

Table 5 Results for the CTOP on the third test set

Instance					*VNS*			*Tabu Feasible*			*Tabu Admissible*		
Instance	*n*	*m*	*Q*	T_{max}	*p*	*CPU*	*%*	*p*	*CPU*	*%*	*p*	*CPU*	*%*
3	101	2	200	200	**536**	305	0.00	535	155	0.19	**534**	30	0.37
3	101	3	200	200	**762**	643	0.00	761	146	0.13	758	48	0.53
3	101	4	200	200	**950**	961	0.00	946	110	0.42	947	42	0.32
6	51	2	160	200	**403**	27	0.00	**403**	9	0.00	**403**	5	0.00
6	51	3	160	200	**565**	43	0.00	**565**	6	0.00	**565**	6	0.00
6	51	4	160	200	**683**	53	0.00	**683**	5	0.00	682	4	0.15
7	76	2	140	160	**377**	107	0.00	376	51	0.27	376	53	0.27
7	76	3	140	160	**548**	164	0.00	547	59	0.18	544	42	0.74
7	76	4	140	160	**707**	296	0.00	**707**	44	0.00	702	39	0.71
8	101	2	200	230	**536**	211	0.00	534	86	0.37	533	42	0.56
8	101	3	200	230	**762**	482	0.00	761	81	0.13	**762**	29	0.00
8	101	4	200	230	**950**	726	0.00	949	89	0.11	949	38	0.11
9	151	2	200	200	**547**	952	0.00	**547**	603	0.00	546	289	0.18
9	151	3	200	200	**796**	1805	0.00	794	606	0.25	794	188	0.25
9	151	4	200	200	**1033**	2903	0.00	**1033**	480	0.00	1029	254	0.39
10	200	2	200	200	**556**	2214	0.00	554	899	0.36	555	825	0.18
10	200	3	200	200	**815**	3467	0.00	814	1152	0.12	814	858	0.12
10	200	4	200	200	**1064**	–	0.00	**1064**	1260	0.00	1063	789	0.09
13	121	2	200	720	**513**	253	0.00	512	102	0.20	512	70	0.20
13	121	3	200	720	**727**	557	0.00	**724**	96	0.41	724	67	0.41
13	121	4	200	720	**908**	954	0.00	907	76	0.11	906	40	0.22
14	101	2	200	1040	**534**	109	0.00	531	51	0.56	**534**	60	0.00
14	101	3	200	1040	**770**	210	0.00	750	54	2.67	**770**	41	0.00
14	101	4	200	1040	**975**	483	0.00	**975**	56	0.00	**975**	38	0.00
15	151	2	200	200	**550**	929	0.00	549	523	0.18	**550**	278	0.00
15	151	3	200	200	**801**	1711	0.00	**801**	629	0.00	800	327	0.13
15	151	4	200	200	**1031**	2832	0.00	1019	618	1.18	1030	276	0.10
16	200	2	200	200	**558**	2040	0.00	556	1342	0.36	557	1099	0.18
16	200	3	200	200	**821**	3360	0.00	818	988	0.37	820	1036	0.12
16	200	4	200	200	**1073**	–	0.00	1072	1263	0.09	1071	897	0.19
Average							0.00			0.29			0.22

Table 6 Results for the CPTP on the third test set

Instance				*VNS*			*Tabu Feasible*			*Tabu Admissible*		
Instance	*n*	*m*	*Q*	*v*	*CPU*	*%*	*v*	*CPU*	*%*	*v*	*CPU*	*%*
3	102	2	200	**330.14**	158	0.00	319.28	39	3.40	319.28	36	3.40
3	102	3	200	**447.15**	254	0.00	444.82	39	0.52	433.38	69	3.18
3	102	4	200	536.64	403	0.19	**537.66**	45	0.00	536.13	36	0.29
6	52	2	160	**168.60**	14	0.00	**168.60**	5	0.00	**168.60**	4	0.00
6	52	3	160	**219.36**	23	0.00	218.96	5	0.18	218.96	2	0.18
6	52	4	160	**258.97**	23	0.00	**258.97**	6	0.00	254.47	4	1.77
7	77	2	140	**199.97**	52	0.00	**199.97**	25	0.00	**199.97**	52	0.00
7	77	3	140	**274.80**	85	0.00	**274.80**	33	0.00	**274.80**	40	0.00
7	77	4	140	**344.35**	120	0.00	343.12	35	0.36	339.95	51	1.30
8	102	2	200	**330.14**	158	0.00	319.28	39	3.40	319.28	36	3.40
8	102	3	200	**447.15**	251	0.00	444.82	39	0.52	433.38	69	3.18
8	102	4	200	536.64	403	0.19	**537.66**	45	0.00	536.13	36	0.29
9	152	2	200	**347.90**	548	0.00	347.43	158	0.14	**347.90**	278	0.00
9	152	3	200	**500.17**	902	0.00	496.84	135	0.67	500.12	323	0.01
9	152	4	200	**639.72**	1770	0.00	635.67	178	0.64	633.64	238	0.96
10	201	2	200	**382.41**	1433	0.00	378.32	500	1.08	379.81	984	0.68
10	201	3	200	**559.80**	2110	0.00	549.83	504	1.81	551.44	710	1.51
10	201	4	200	**723.47**	–	0.00	710.59	348	1.81	719.13	707	0.60
13	122	2	200	**239.57**	183	0.00	238.58	64	0.41	230.59	79	3.89
13	122	3	200	**250.69**	227	0.00	234.99	62	6.68	244.96	121	2.34

Table 6 Continued

Instance				***VNS***			***Tabu Feasible***			***Tabu Admissible***		
Instance	*n*	*m*	*Q*	*v*	*CPU*	*%*	*v*	*CPU*	*%*	*v*	*CPU*	*%*
13	122	4	200	279.43	262	5.38	264.46	61	11.34	**294.46**	125	0.00
14	102	2	200	**303.17**	138	0.00	302.94	52	0.08	**303.17**	80	0.00
14	102	3	200	**418.28**	222	0.00	416.32	59	0.47	417.32	84	0.23
14	102	4	200	**537.24**	540	0.00	516.20	58	4.08	531.53	99	1.08
15	152	2	200	**378.09**	575	0.00	**378.09**	142	0.00	**378.09**	259	0.00
15	152	3	200	**519.39**	768	0.00	517.18	209	0.43	512.83	301	1.28
15	152	4	200	653.2	1364	0.26	**654.94**	177	0.00	652.58	285	0.36
16	201	2	200	**394.05**	1354	0.00	390.47	381	0.92	391.71	876	0.60
16	201	3	200	**567.24**	1985	0.00	558.61	483	1.55	558.10	985	1.64
16	201	4	200	729.40	3572	0.24	**731.14**	352	0.00	726.22	809	0.68
Average						0.21			1.35			1.09

algorithms because the exact branch-and-price algorithm is not able to solve such large instances. The results are shown in Table 5 for the CTOP and in Table 6 for the CPTP.

For the CTOP (Table 5), *VNS* has produced all best known solutions, while *TabuFeasible* and *TabuAdmissible* find nine and seven best known solutions, respectively. Notice, however, that the three heuristics produce results of similar quality since the average deviation from the best known solution values is less than 0.3% for both tabu search algorithms. With the exception of two instances (namely, instance 14 with $m = 3$ and instance 15 with $m = 4$), all deviations are smaller than 1.0%. For the CPTP (Table 6), there is also a small advantage for *VNS* that produces 25 best known solutions, to be compared with the nine and seven best known solutions found by *TabuFeasible* and *TabuAdmissible*, respectively. In this case, the deviation from the best known solution values can raise up to 11%, but it is less than 2%, in average, for all algorithms.

6. Conclusions

We have developed one exact and three heuristic algorithms for the capacitated versions of two VRPs with profits, namely the CTOP and the CPTP. The proposed algorithms are extensions of recent algorithms developed for the same problems without the capacity constraints. The computational experiments show that the heuristic procedures often find the optimal solution of instances that can be solved to optimality by a branch-and-price algorithm. For more difficult instances, for which no optimal solution can be generated within a reasonable time limit, the three heuristic methods produce similar results, since they deviate one from the other by less than 2% in average.

References

Archetti C, Hertz A and Speranza MG (2007). Metaheuristics for the team orienteering problem. *J Heuristics* **13**: 49–76.

Boussier S, Feillet D and Gendreau M (2007). An exact algorithm for team orienteering problems. *4 OR* **5**: 211–230.

Butt SE and Cavalier TM (1994). A heuristic for the multiple tour maximum collection problem. *Comput Opns Res* **21**: 101–111.

Chao I-M, Golden B and Wasil EA (1996). The team orienteering problem. *Eur J Opl Res* **88**: 464–474.

Christofides N, Mingozzi A and Toth P (1979). The vehicle routing problem. In: Christofides N, Mingozzi A, Sandi C and Toth P (eds). *Combinatorial Optimization*. Wiley: Chichester. pp 315–338.

Desaulniers G, Desrosiers J and Solomon MM (eds) (2005). *Column generation, GERAD 25th Anniversary Series*. Springer: New York.

Feillet D, Dejax P and Gendreau M (2005). Traveling salesman problems with profits. *Transport Sci* **39**: 188–205.

Feillet D, Dejax P, Gendreau M and Gueguen C (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problem. *Networks* **44**: 216–229.

Feillet D, Gendreau M, Rousseau LM. (2005). *New Refinements for the solution of vehicle routing problems with branch and price*. *Technical report* CRT-2005-08. Centre de Recherche sur les Transports, Montreal, Canada.

Gendreau M, Hertz A and Laporte G (1994). A tabu search heuristic for the vehicle routing problem. *Mngt Sci* **40**: 1276–1290.

Mladenović N and Hansen P (1997). Variable neighborhood search. *Comput Opns Res* **24**: 1097–1100.

Tang H and Miller-Hooks E (2005). A tabu search heuristic for the team orienteering problem. *Comput Opns Res* **32**: 1379–1407.

Toth P and Vigo D. (eds) (2002). *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications: Philadelphia, USA.

Received April 2007;
accepted February 2008 after one revision