Target aiming Pareto search and its application to the vehicle routing problem with route balancing

Article *in* Journal of Heuristics · September 2007

DOI: 10.1007/s10732-007-9022-6 · Source: DBLP

3 authors:

Nicolas Jozefowiez
University of Lorraine

Frédéric Semet
École Centrale de Lille

El-Ghazali Talbi
University of Lille Nord de France

J Heuristics (2007) 13: 455–469
DOI 10.1007/s10732-007-9022-6

Target aiming Pareto search and its application to the vehicle routing problem with route balancing

Nicolas Jozefowiez · Frédéric Semet · El-Ghazali Talbi

Received: 17 October 2003 / Revised: 12 July 2006 / Accepted: 7 August 2006 / Published online: 5 April 2007

Abstract In this paper, we present a solution method for a bi-objective vehicle routing problem, called the vehicle routing problem with route balancing (VRPRB), in which the total length and balance of the route lengths are the objectives under consideration. The method, called Target Aiming Pareto Search, is defined to hybridize a multi-objective genetic algorithm for the VRPRB using local searches. The method is based on repeated local searches with their own appropriate goals. We also propose an implementation of the Target Aiming Pareto Search using tabu searches, which are efficient meta-heuristics for the vehicle routing problem. Assessments with standard metrics on classical benchmarks demonstrate the importance of hybridization as well as the efficiency of the Target Aiming Pareto Search.

Keywords Routing · Multi-objective optimization · Tabu search · Hybrid algorithm

1 Introduction

This paper investigates the solution of a multi-objective combinatorial optimization (MOCO) problem by means of local search heuristics (LS). LS may be used as stand alone solvers (Serafini 1992; Ulungu 1993; Hansen 1997; Knowles 2002) or for hybridization, a methodology that uses an initial heuristic to generate an approximation,

N. Jozefowiez (✉) · E.-G. Talbi
Laboratoire d'Informatique Fondamentale de Lille, Université des Sciences et Technologies de Lille, Villeneuve d'Ascq, France
e-mail: jozefowiez@fulbrightweb.org

E.-G. Talbi
e-mail: talbi@lifl.fr

F. Semet
Laboratoire d'Automatique, de Mécanique et d'Informatique industrielles et Humaines, Université de Valenciennes et du Hainaut-Cambrésis, Valenciennes, France
e-mail: frederic.semet@univ-valenciennes.fr

Table 1 Objective values for the best found solutions of Taburoute and Prins' GA

Instance	Taburoute		Prins' genetic algorithm	
	Distance	Balance	Distance	Balance
E51-05e	524.61	20.07	524.61	20.07
E76-10e	835.32	78.10	835.26	91.08
E101-08e	826.14	97.88	826.14	97.88
E151-12c	1031.17	98.24	1031.63	100.34
E200-17c	1311.35	106.70	1300.23	82.31
E121-07c	1042.12	146.67	1042.12	146.67
E101-10c	819.56	93.43	819.56	93.43

which is then improved through the application of a second algorithm (Ben Abdelaziz et al. 1999; Deb and Goel 2001; Jaszkiewicz 2002; Ishibuchi et al. 2002). (More details and references can be found in (Ehrgott and Gandibleux 2000).) We introduce a method that uses LS designed to hybridize a genetic algorithm for a bi-objective vehicle routing problem (VRP).

The elementary version of the vehicle routing problem is the capacitated vehicle routing problem (CVRP). The CVRP can be modeled on a complete graph whose respective vertices are associated with n customers and to the depot. Each customer must be served a quantity q_i of goods ($i = 1, \dots, n$) from a unique depot. Vehicles are available to deliver those goods, and each vehicle has a maximal amount Q of goods that it can transport. A solution for the CVRP involves a collection of routes in which each customer is visited only once, and the total demand for each route is, at most, Q. The CVRP aims to determine a minimal total length solution. This problem has been proved NP-hard Lenstra and Rinnooy Kan (1981), and solution methods range from exact methods to specific heuristic and meta-heuristic approaches Toth and Vigo (2001).

In this paper, we address a bi-objective extension of the CVRP: the vehicle routing problem with route balancing (VRPRB). This problem was first introduced in (Jozefowiez et al. 2002). The objectives are the minimization of:

1. The distance traveled by the vehicles.
2. The difference between the longest route length and the shortest route length.

Table 1 presents the values found for these objectives using Taburoute (Gendreau et al. 1994) and Prins' genetic algorithm (Prins 2006) for the seven CVRP benchmarks proposed by Christofides et al. (1979). The best-known solutions currently available for the CVRP are of poor quality when the route balancing objective is considered.

The application of multi-objective optimization to the VRP addresses two primary goals. First, such optimization can be used to adapt the problem to a specific application (Current and Marsh 1993; Corberan et al. 2002; El-Sherbeny 2001) or to improve the practical aspects of the model (Ribeiro and Loureno 2001; Lee and Ueng 1999). Second, it can be used to tackle the difficulties introduced by the constraints of the vehicle routing problem with time windows (Sessomboon et al. 1998; Hong 1999; Geiger 2001; Rahoual et al. 2001). Generally, the above authors allow the violation of the time windows by incorporating objectives that minimize the number of violations and the earliness and lateness of the vehicle relative to the time window bounds.

In most researches dealing with the VRP, the total length of the solution is minimized. However Corberan et al. (2002) disregard the solution length in favor of minimizing the number of vehicles, allowing the particularity of the context to be taken into account, i.e. the organization of a school bus network in a rural area of Spain. Although minimizing the number of vehicles is also an objective in (Sessomboon et al. 1998; Geiger 2001; Rahoual et al. 2001), the authors still minimize the total length of the solution.

Another frequently considered objective is the balance of the routes. In (Lee and Ueng 1999), the authors balance the time needed for each trip, expressing it as the sum of the differences between each route length and the shortest route length. Route balancing is also an objective for Ribeiro and Loureno (2001) who address a three objective multi-period vehicle routing problem. In their paper the balance is measured by the standard deviation of the load of the routes, which consists of the number of customers visited. Corberan et al. (2002) work on minimizing the time spent in a bus, which can also be viewed as a type of route balancing. One of the eight objectives described by El-Sherbeny (2001) is the same as our second objective: the minimization of the difference between the maximal route length and the minimal route length.

Earlier studies of multi-objective VRP (Current and Marsh 1993) used exact algorithms, mathematical methods like goal programming, and specific exchange/insertion heuristics. Several meta-heuristics have also been applied, including scatter search (Corberan et al. 2002), iterated local search (Ribeiro and Loureno 2001), simulated annealing (El-Sherbeny 2001), and genetic algorithms (Geiger 2001; Rahoual et al. 2001). The hybridization of a genetic algorithm using a local search has also been proposed (Sessomboon et al. 1998). Generally, the use of meta-heuristics is motivated by the intractability of the problem, while the use of LS, especially tabu searches (TS), is motivated by the good results obtained using such methods on the VRP (Toth and Vigo 2001).

The present paper is organized as follows: Sect. 2 introduces our Target Aiming Pareto Search (TAPaS) methodology. Section 3 provides an implementation of TAPaS for the VRPRB, as well as an implementation of NSGA II (Deb et al. 2002) as used for the approach assessment. In Sect. 4, we assess the efficiency of TAPaS on a classic testbed using standard metrics. Section 5 presents our conclusions.

2 A new method for solving multi-objective problems

2.1 Solution of a multi-objective problem

A multi-objective problem can be stated as follows:

$$(\text{MOP}) = \begin{cases} \min F(x) = (f_1(x), f_2(x), \ldots, f_n(x)) \\ \text{s.t. } x \in D \end{cases} \tag{1}$$

with $n \geq 2$ being the number of objective functions; $x = (x_1, x_2, \ldots, x_r)$, the decision variable vector; D, the feasible solution space; and $F(x)$, the objective vector. The set $O = F(D)$ corresponds to the feasible solutions in the objective space, and $y = (y_1, y_2, \ldots, y_n)$, where $y_i = f_i(x)$, is a solution. A MOP solution is the set of the non-dominated solutions called the Pareto set (PS). Dominance is defined as follows:

Definition 2.1 A solution $y = (y_1, y_2, \dots, y_n)$ dominates (denoted $\prec$) a solution $z = (z_1, z_2, \dots, z_n)$ if and only if $\forall i \in \{1 \dots n\}$, $y_i \leq z_i$ and $\exists i \in \{1 \dots n\}$, such that $y_i < z_i$.

Definition 2.2 A solution y found by an algorithm A is said to be potentially Pareto optimal (PPS), relative to A, if A does not find a solution z, such that z dominates y.

Evolutionary algorithms and local search methods have generally been proposed to approximate PS (Ehrgott and Gandibleux 2000). Such heuristics must be designed with two goals in mind: (i) the algorithm must converge toward the PS, and (ii) the identified solutions must be well-diversified along the frontier.

2.2 Target aiming Pareto search

As pointed out by Ehrgott and Gandibleux (2000), LS have to be guided in their approximation of the PS. In our methodology, this is done via an iterative process based on cooperative LS and on appropriate goals.

In TAPaS, a local search heuristic l_i is applied to each solution s_i of a potentially Pareto set P. A specific objective function o_i is defined for each local search l_i. The function o_i must take into account the multiplicity of the LS invoked. Indeed, two LS should not examine the same area of the objective space, and the entire area that dominates P should be explored in order to converge toward the optimal PS. The definition of o_i is based on the partition of O according to P (Fig. 1):

- $A_d = \{s \in O | \exists s' \in P, s' \prec s\}$
- $A_{\text{nd}} = \{s \in O | \forall s' \in P, (s' \not\prec s) \wedge (s \not\prec s')\}$
- $A_s = \{s \in O | \exists! s' \in P, s \prec s'\}$
- $A_p = \{s \in O | \exists s_1, s_2 \in P, (s \prec s_1) \wedge (s \prec s_2)\}$

Each solution $s_i \in P$ is associated with a part A_s^i of A_s. If l_i is able to generate a feasible solution in A_s^i, then the approximation is improved according to the convergence, without impoverishing the diversification.

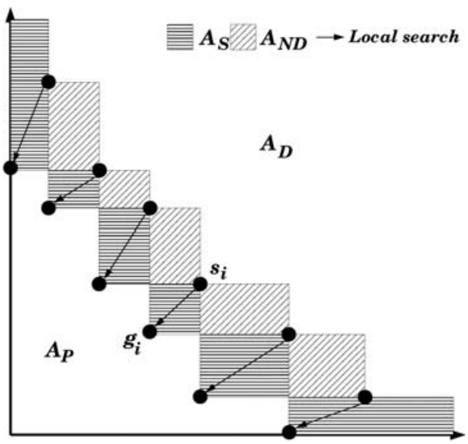


Fig. 1 Partition of O

To guide the search, a goal g_i is given to each l_i, with g_i being the point that dominates all points of A_s^i. In cases where certain coordinates of g_i cannot be defined (e.g. the extremities of P), a lower bound for the missing coordinates should be used. Algorithm 1 computes the coordinates of g_i.

Then, o_i is stated as follows:

$$\min\left(\sum_{j=1}^{M} |f_j(s) - f_j(g_i)|^r\right)^{\frac{1}{r}}. \tag{2}$$

The general loop of a LS is described in Algorithm 2. When l_i reaches g_i or when it finds a solution that dominates g_i, it stops and produces an archive a_i which contains all the current solutions that are not dominated.

When all l_i are terminated, a new set P' is formed by the Pareto union of all a_i. Because P' might be improved by another application of LS, the complete process is iterated until P' does not differ from P. At each iteration, a clustering algorithm may be applied to P in order to insure that the starting solutions are not too close and that the g_i are more relevant. Moreover, applying the clustering algorithm may reduce the computational time required. This loop forms the core of TAPaS, as shown in Algorithm 3.

Algorithm 1 goal_point(s: solution, P: Pareto set)

{M is the number of objectives}
for all objective m **do**
  $f_m(g_p) \leftarrow \arg\min_{\{f_m(s')|(s' \in P) \wedge (f_m(s') < f_m(s))\}}(f_m(s') - f_m(s))$
end for
return g_p

Algorithm 2 local_search(s: solution, g_p: point, L: local search)

{M is the number of objectives}
{$N(s)$ is the neighborhood of the solution s}
$L.s \leftarrow s$
$L.a \leftarrow \{s\}$
$L.o \leftarrow \min(\sum_{j=1}^{M} |f_j(s) - f_j(g_p)|^r)^{\frac{1}{r}}$
while $(L.s \not\prec g_p) \wedge \neg$ *L.stopping_criterion* **do**
  $s_{inter} \leftarrow \{s^* \in N(L.s) | \forall s' \in N(L.s), L.o(s^*) \leq L.o(s')\}$
  $L.s \leftarrow s_{inter}$
  $L.a \leftarrow L.a \setminus \{s_a \in L.a | L.s \prec s_a\}$
  if $\forall s_a \in L.a, s_a \not\prec L.s$ **then**
    $L.a \leftarrow L.a \cup \{L.s\}$
  end if
end while
return $L.a$

Algorithm 3 TAPaS(P: Pareto set, L: local search)

```
t ← 0
P_t ← P
continue ← true
while continue do
    P_{t+1} ← P_t
    cluster(P_t)
    for all s ∈ P_t do
        g_p ← goal_point(s, P_t)
        a ← local_search(s, g_p, L)
        P_{t+1} ← P_{t+1} ∪ a
    end for
    if P_t ≠ P_{t+1} then
        t ← t + 1
    else
        continue ← false
    end if
end while
return P_t
```

In this context, the term local search is used in a general sense; it can also refer to a Tabu search approach or a simulated annealing algorithm, for example.

3 Implementation

3.1 Tabu search

In our implementation of TAPaS for the VRPRB, the LS is a tabu search (TS) based on the Unified Tabu Search (UTS) (Cordeau et al. 1997; Cordeau et al. 2001). UTS was initially designed for VRP with time windows, but it also performs well on CVRP. UTS has many interesting features:

1. UTS allows infeasible solutions to be explored during the search, which means that a solution s that does not respect capacity constraints will be visited, but its cost $c(s)$ will deteriorate according to a specific rule. Let $q(s)$ denote the total violation of capacity constraints. The cost of s becomes $f(s) = c(s) + \alpha q(s)$, where α is a positive parameter that is dynamically adjusted.
2. In UTS, the neighborhood of a solution s is defined using an attribute set $B(s) = \{(i, k) : \text{customer } i \text{ is visited by vehicle } k\}$. The neighborhood $N(s)$ is defined by removing an attribute (i, k) from $B(s)$ and replacing it with another attribute (i, k'). When a customer i is removed from a route k, route k is repaired by linking the customers before and after customer i, while adding i to k' to minimize $f(s)$. When a customer i is removed from route k, the attribute (i, k) is marked tabu for the θ following iterations.

3. UTS also proposes an aspiration criterion that allows the tabu status of an attribute to be revoked. However, since the number of iterations performed in our implementation is limited, and since the aspiration criterion tends to slow down the search during the initial iterations, we did not implement it.
4. To diversify the UTS search, any solution $s' \in N(s)$, such that $f(s') \geq f(s)$, is penalized. Let ρ_{ik} be the number of times the attribute (i,k) has been added to the solution during the search process. Then a penalty $p(s') = \lambda c(s')\sqrt{nm}\sum_{(i,k)\in B(s')}\rho_{ik}$, where λ is a positive constant, is added to $f(s')$.
5. UTS only needs one parameter: the maximum number of iterations was fixed to 5000. This value was fixed by experimentation.

To consider the presence of two objectives, our implementation differs from the standard UTS by incorporating two diversity penalties, which can be defined generally as: $p(s) = \lambda r(s)\sqrt{nm}\sum_{(i,k)\in B(s')}\rho_{ik}$. The first penalty, denoted p_l, is length-oriented, and requires, for a solution s, $r(s)$ to be equal to the length of the solution. The second one, denoted p_b, is balance-oriented, and requires $r(s)$ to be equal to the value of the second objective. The choice of penalty for a given TS depends on the position of the starting solution in the current potentially Pareto set. In our implementation, p_l was attributed to the TS starting at solution s with the following probability:

$$\frac{1}{2} \times \left(1 - \frac{f_1(s) - f_1^{\min}}{f_1^{\max} - f_1^{\min}}\right) + \frac{1}{2} \times \frac{f_2(s) - f_2^{\min}}{f_2^{\max} - f_2^{\min}}, \tag{3}$$

where $f_1^{\min}$ (resp. $f_2^{\min}$) is the current best found value for the first objective (resp. the second objective), and $f_1^{\max}$ (resp. $f_2^{\max}$) the worst value of the current potentially Pareto set. The penalty was introduced to help UTS escape from local optima for the solutions that were near the best found solution for the balance objective and for which a diversity penalty based on the length was not always relevant.

During the clustering phase, the *average linking method* (Morse 1980) was applied to the current approximation, allowing up to 10 solutions to be chosen. This value has been chosen experimentally and by considering the spread of the efficient set as well as the computational time needed for an iteration of the Tabu search. It is clear that this value strongly depends on the problem to be solved and the used techniques.

3.2 NSGA II

To assess the efficiency of our methodology and to provide starting sets for the hybridization process, we also implemented NSGA II (Deb et al. 2002) for the VRPRB. Algorithm 4 outlines the main loop of NSGA II. This section outlines our adaptations of NSGA II. NSGA II sorts the population into different non-domination levels. In this ranking phase, the non-dominated individuals in the population obtain rank 1 and form the subset E_1. Rank k is given to the solutions only dominated by the individuals belonging to the subset $E_1 \cup E_2 \cup \cdots \cup E_{k-1}$. Then, a fitness equal to its rank (1 is the best level) is assigned to each solution.

NSGA II also uses a *crowding distance metric* to provide diversity during the search. This metric gives an estimate of the density of the solutions surrounding a solution i in the population. This estimate is expressed by approximating the perimeter of the cuboid formed by the nearest neighbors of i. Then, a binary tournament is used to select parents from the current population to generate offsprings. During the tournament, two solutions are compared by means of a *crowded tournament selection operator*. According to this operator, a solution i wins a tournament against another solution j if any of the following conditions is true:

1. $r_i < r_j$.
2. $r_i = r_j$ and $d_i > d_j$, where d_i is the crowding distance of solution i.

The first condition makes sure that the chosen solution lies in a better non-dominated set. The second condition breaks ties between two solutions belonging to the same non-dominated front by considering their crowding distance, in other words, by selecting the less crowded individual.

We added an archive a whose purpose is to store the non-dominated solutions as they are found. Doing so insures that no non-dominated solutions will be lost due to the stochasticity of the algorithm. Algorithm 4 was run until it was not able to improve the archive for 5000 generations, and the size of each population, denoted N, was fixed to 128. These values were shown to be effective by preliminary experimentation.

Algorithm 5 outlines the recombination phase. The following VRP operators were used: the crossover RBX (Potvin and Bengio 1996), the crossover SPLIT inspired by Prins' GA (Prins 2006), and the mutation operator Or-opt (Or 1976). A 2-opt local search (Lin 1965) on the routes was also applied.

Algorithm 4 NSGA II

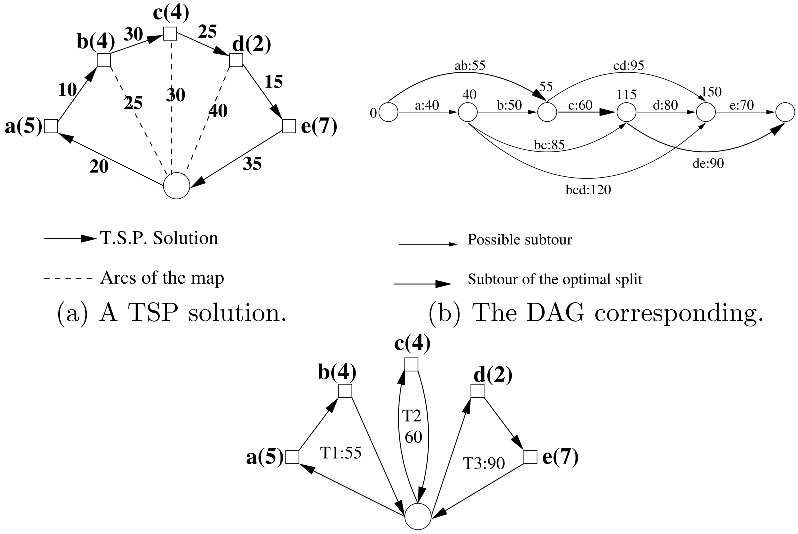
$t \leftarrow 0$
$P_t \leftarrow initial\_population()$
$Q_t \leftarrow initial\_population()$
$R_t \leftarrow P_t \cup Q_t$
$a \leftarrow \{s \in R_t | \forall s' \in R_t, s' \nprec s\}$
while $\neg stopping\_criterion$ **do**
  $non\_dominated\_sort(P_t)$
  $P_{t+1} \leftarrow \emptyset$
  $i \leftarrow 1$
  while $|P_{t+1}| + |\mathcal{F}_i| < N$ **do**
    $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i$
    $i \leftarrow i + 1$
  end while
  $crowding\_sort(\mathcal{F}_i)$
  Include the $N - |P_{t+1}|$ most widely spread solutions of $\mathcal{F}_i$ to P_{t+1}
  $recombination_phase(P_{t+1}, Q_{t+1})$
  $a \leftarrow a \setminus \{s \in a | \exists s' \in Q_{t+1}, s' \prec s\} \cup \{s \in Q_{t+1} | \forall s' \in a, s' \nprec s\}$
  $t \leftarrow t + 1$
end while
return a

We chose the route based crossover (RBX) (Potvin and Bengio 1996) in order to emphasize the balancing criterion. This operator retains several randomly chosen first-parent routes, and completes the offspring using routes from the second parent. Customers that have already been used must be removed from the routes provided by the second parent.

The SPLIT crossover, based on the genetic algorithm proposed by Prins (2006), treats parents as two different traveling salesman problem (TSP) solutions. The arcs entering or leaving the depot are removed and the routes are merged. The OX crossover is applied, and a VRP solution is built from the offspring. This is done by building a directed acyclic graph (DAG) $D = (V, A)$ with $V = \{0, 1, \ldots, m\}$ and by identifying the shortest path within it. Let the permutation σ represent the TSP solution, where $\sigma(0)$ still equals 0. Different feasible routes (i.e. the split) are represented by the arc of D, and for all $i, j \in V$ $(i < j)$, the arc (i, j) belongs to A if and only if the total load on the route $0\sigma(i+1)\ldots\sigma(j)0$ is less or equal to the vehicle capacity. The valuation of the arc (i, j) is calculated with the formula: $d(0, \sigma(i+1)) + \sum_{l=0}^{j-1} d(\sigma(i+l), \sigma(i+l+1)) + d(\sigma(j), 0)$, where $d(c_1, c_2)$ is the distance between customers c_1 and c_2. Finding the optimal partition of the TSP tour is equivalent to solving a shortest path problem on the DAG. An example provided by Prins (2006) is presented in Fig. 2.

Algorithm 5 recombination_phase(P, Q: POPULATION)

```
Q ← ∅
for i ← 1, ..., N do
  pa_1 ← choose_solution(P)
  s ← choose_solution(P)
  if s >_c pa_1 then
    pa_1 ← s
  end if
  pa_2 ← choose_solution(P)
  s ← choose_solution(P)
  if s >_c pa_2 then
    pa_2 ← s
  end if
  if rand() < 0.5 then
    s ← RBX(pa_1, pa_2)
  else
    s ← SPLIT(pa_1, pa_2)
  end if
  if rand() < 0.4 then
    s ← or_opt(s)
  end if
  ls_2opt(s)
  Q ← Q ∪ {s}
end for
```



(a) A TSP solution.
(b) The DAG corresponding.
(c) The VRP solution corresponding to the shortest path in the DAG of (b).

Fig. 2 From a TSP solution to a VRP solution

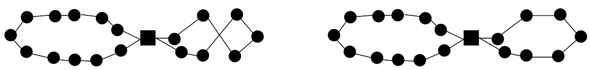

(a) (b)

Fig. 3 **(a)** is better-balanced than **(b)**, but **(b)** does not artificially improve the balance

The Or-opt mutation operator (Or 1976) moves 1 to K consecutive customers from one route to another. In our implementation, K was set at 3. This operator has two advantages: (i) it is particularly well-adapted to the VRP solution, which is a set of routes, and (ii) it allows new routes to be created.

Finally, a 2-opt local search (Lin 1965) is applied to each route of each offspring. Using this kind of TSP improvement procedure achieves three purposes: (i) it makes the solution less chaotic; (ii) it improves the total length; and (iii) it prevents the balancing objective from being distorted, as shown in Fig. 3.

4 Computational results

TAPaS and NSGA II for the VRPRB were both coded in C. The evaluation was conducted on seven standard Christofides benchmarks (Christofides et al. 1979), which correspond to CVRP instances, as well as on eight Taillard instances (Rochat and Taillard 1995). The number of customers varies from 50 to 199. Ten runs were made for each instance. As suggested in (Knowles and Corne 2002), the S metric (Zitzler 1999) was used. $S(A)$ represents the size of the area dominated by the approximation generated by an algorithm A. It is based on computing the volume (area in the bi-objective case) dominated by a given Pareto-front approximation. The S metric

Table 2 Average computational times for the hybrid meta-heuristic (in seconds)

Instance	NSGA II	TAPaS	Hybrid meta-heuristic
E51-05e	314	78	392
E76-10e	1576	348	1925
E101-08e	2594	895	3489
E151-12c	4438	2282	6520
E200-17c	5708	4428	10136
E121-07c	4648	1488	6136
E101-10c	2767	682	3449
Tai100a	2266	1051	3317
Tai100b	2589	1152	3741
Tai100c	3180	1339	4519
Tai100d	3010	1186	4519
Tai150a	4312	2157	6469
Tai150b	4600	2920	7520
Tai150c	4652	2186	6838
Tai150d	4459	2112	6571

requires a reference point Z_{ref} consisting of a reference value for each of the two objectives. For both objectives and for each instance, we used the worst value found by all methods. We have normalized the S metric. We also used the C metric (Zitzler 1999), where $C(A, B)$ gives the ratio of the approximation generated by B, which is dominated by the approximation generated by A. First, we have assessed the efficiency of TAPaS. To do that, we solved the problems by the means of NSGA II such as it has been explained in Sect. 3.2. Then, TAPaS is applied on the approximation generated by NSGA II. The combination of NSGA II and TAPaS forms a hybrid meta-heuristic (HM). In Table 2, we report the average times for each part of HM and the total average time. Then, the instances were solved using NSGA II assuring that NSGA II runs at least the average computational time required by the HM, i.e. the sum of the average times of NSGA II and TAPaS. These runs of NSGA II are denoted as NSGAu II in the remaining of the paper. Table 3 provides the maximum, mean, and minimum S metric values for HM and NSGAu II.

These results show that the worst run of the HM is better on nine over 15 instances than the average value for NSGAu II. Moreover it is always better than the worst values of NSGAu II. The average value for the HM is always better than the average value for NSGAu II and it is better on four instances than the best run of NSGAu II. Finally, the HM is able on average to find a better approximation than NSGAu II on 14 out of 15 instances. All in all, it appears that the HM is significantly better than NSGAu II on the set of instances notably on the largest ones.

We have also assessed the contribution of TAPaS over NSGA II by computing the ratio of solutions improved by the application of TAPaS on the approximations of NSGA II. Table 4 reports the maximum, mean, and minimum values as well as the standard deviation for the metric C(HM, NSGA II). On the Christofides' instance, except for benchmarks E51-05e and E101-10c, which correspond to the easiest instances of the testbed and for which NSGA II was already able to provide good quality sets, TAPaS is able to improve, on average, more than 50 percent of the approxi-

Table 3 Comparison between the hybrid meta-heuristic and NSGAu II (using the $\mathcal{S}$ metric)

	HM			NSGAu II		
	max.	mean	min.	max.	mean	min.
E51-05e	0.184722	0.181876	0.178224	0.183291	0.179356	0.173523
E76-10e	0.508462	0.504729	0.501610	0.509578	0.501229	0.495054
E101-08e	0.505894	0.504372	0.502449	0.503396	0.502078	0.500110
E151-12c	0.582746	0.576487	0.573160	0.577029	0.571535	0.559313
E200-17c	0.574079	0.567778	0.559463	0.560310	0.547665	0.539729
E121-07c	0.475286	0.474632	0.473764	0.474161	0.471848	0.468537
E101-10c	0.574618	0.558285	0.542310	0.571528	0.558178	0.543720
Tai100a	0.515336	0.507331	0.500008	0.503306	0.500602	0.496800
Tai100b	0.556439	0.549141	0.543534	0.551838	0.548910	0.544449
Tai100c	0.567101	0.564852	0.562087	0.566791	0.562932	0.559125
Tai100d	0.557676	0.554981	0.550859	0.554323	0.549609	0.544318
Tai150a	0.582285	0.573663	0.566718	0.571384	0.561065	0.549111
Tai150b	0.618611	0.612231	0.604337	0.614213	0.608815	0.597530
Tai150c	0.632052	0.627812	0.623626	0.629449	0.617879	0.599805
Tai150d	0.612510	0.599726	0.593876	0.594941	0.590824	0.586338

Table 4 Ratio of improved solutions when TAPaS is applied (using C(HM, NSGA II))

Instance	max.	mean	min.	std. dev.
E51-05e	0.90	0.47	0.11	0.30
E76-10e	0.79	0.51	0.19	0.26
E101-08c	1.00	0.89	0.83	0.06
E151-12c	0.97	0.85	0.75	0.08
E200-17c	1.00	0.93	0.74	0.10
E121-07c	0.91	0.70	0.53	0.13
E101-10c	0.54	0.43	0.19	0.13
Tai100a	0.99	0.93	0.87	0.06
Tai100b	1.00	0.82	0.40	0.21
Tai100c	0.79	0.62	0.50	0.11
Tai100d	0.96	0.86	0.56	0.15
Tai150a	0.99	0.79	0.64	0.11
Tai150b	0.97	0.88	0.71	0.10
Tai150c	0.92	0.81	0.69	0.07
Tai150d	0.95	0.85	0.77	0.06

mations. Furthermore, the improvement is clearly significant for the largest instances. The ratio of improved solutions is also very important on the complete Taillard set of instances.

Finally, Table 5 provides the maximum, mean, and minimum S metric values when TAPaS is applied on the non-dominated solutions obtained from a randomly generated population instead of an approximation generated by NSGA II. These values

Table 5 TAPaS applied on randomly generated populations (using the $\mathcal{S}$)

Instance	HM	max.	mean	min.
E51-05e	0.181876	0.183214	0.179635	0.176965
E76-10e	0.504729	0.491070	0.483772	0.479411
E101-08c	0.504372	0.503816	0.500826	0.497350
E151-12c	0.576487	0.583605	0.560629	0.548722
E200-17c	0.567778	0.565825	0.525802	0.502104
E121-07c	0.474632	0.469152	0.465739	0.462468
E101-10c	0.558285	0.505070	0.482232	0.464768
Tai100a	0.507331	0.492090	0.463427	0.446285
Tai100b	0.549141	0.542015	0.498966	0.471237
Tai100c	0.564852	0.556762	0.506838	0.392842
Tai100d	0.554981	0.550294	0.506519	0.449803
Tai150a	0.573663	0.516621	0.501110	0.474819
Tai150b	0.612231	0.552367	0.530957	0.516620
Tai150c	0.627812	0.442535	0.415148	0.375577
Tai150d	0.599726	0.572996	0.546108	0.526384

Table 6 Diversification performance of the methods (using the k-distance)

Instance	NSGA II	HM	NSGAu II
E51-05e	0.632493	0.654776	0.584683
E76-10e	0.406861	0.408982	0.411314
E101-08c	0.385763	0.439261	0.398105
E151-12c	0.288893	0.258150	0.261146
E200-17c	0.300588	0.245579	0.339684
E121-07c	0.262204	0.277943	0.283978
E101-10c	0.238037	0.257504	0.257595
Tai100a	0.241133	0.326460	0.253800
Tai100b	0.256708	0.266706	0.272262
Tai100c	0.251910	0.257392	0.255567
Tai100d	0.235605	0.228658	0.234998
Tai150a	0.239144	0.268288	0.223998
Tai150b	0.225721	0.247684	0.223912
Tai150c	0.233970	0.276934	0.213281
Tai150d	0.245319	0.276214	0.239551

show that, on average, the approximations produced through hybridization are better, especially on the largest instances. This clearly illustrates that hybridizations such as HM are better than invoking each component separately.

Finally, we investigated the effect of the TAPaS on the diversification of the generated sets. For that we used the k-distance. It is a density estimation technique proposed by Zitzler et al. (2001) based on kth nearest neighbor method of Silverman (1986). It computes the distance (in this cases the Euclidean distance) to the kth nearest non-dominated solution, and should be interpreted such as the smaller the better.

Here we use $k = 5$ and compute the mean. In Table 6, we report the averages of the mean values for each instance. From these results, it appears that the HM provides better approximation on 7 out of 15 instances that NSGA II. The same remark is true when we compare HM and NSGAu II. Therefore, it seems that the application of TAPaS has neither a positive nor a negative impact on the diversity of the generated approximations.

5 Conclusion

In this paper, we have proposed the Target Aiming Pareto Search methodology as a component of a hybrid method for generating approximations of Pareto sets. TAPaS is based on repeated cooperative local searches with individual appropriate goals. The design of TAPaS was motivated by the need to improve the results obtained by a multi-objective genetic algorithm. The idea was the following one: the genetic algorithm generates well-diversified approximations of the optimal Pareto set and TAPaS improves the convergence toward the optimal Pareto set. The purpose of the methodology was to solve a bi-objective extension of the capacitated vehicle routing problem, called the vehicle routing problem with route balancing, in which the objectives are both the minimization of the total length and the balancing of the routes (the minimization of the difference between the longest route length and the shortest route length). We have proposed an implementation of TAPaS for this bi-objective VRP problem using a tabu algorithm, which is an efficient meta-heuristic approach to the vehicle routing problems. Assessment with standard metrics was conducted on a classic testbed. Computational experiments led to two important conclusions: (i) TAPaS is efficient when it is used for hybridization, and (ii) the hybridization of TAPaS with another algorithm generates better approximations than their individual counterparts.

Acknowledgements This work was partially supported by the Région Nord-Pas-de-Calais and by the European funding program FEDER. This support is gratefully acknowledged. Thanks are also due to the Associated Editor and to the referees for their valuable comments.

References

Ben Abdelaziz, F., Kricen, S., Chaouchi, J.: A hybrid heuristic for multiobjective knapsack problem. In: Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization, pp. 205–212. Kluwer Academic, Dordrecht (1999)

Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (eds.): Combinatorial Optimization, pp. 315–338, Chapter 11. Wiley, Chichester (1979)

Corberan, A., Fernandez, E., Laguna, M., Marti, R.: Heuristic solutions to the problem of routing school buses with multiple objectives. J. Oper. Res. Soc. **53**(4), 427–435 (2002)

Cordeau, J.-F., Gendreau, M., Laporte, G.: A tabu search heuristic for periodic and multi-depot vehicle problems. Networks **30**, 105–119 (1997)

Cordeau, J.-F., Laporte, G., Mercier, A.: A unified tabu search heuristic for vehicle routing problem with time windows. J. Oper. Res. Soc. **52**, 928–932 (2001)

Current, J., Marsh, M.: Multiobjective transportation network design and routing problems: taxonomy and annotation. Eur. J. Oper. Res. **65**, 4–19 (1993)

Deb, K., Agrawal, S., Pratab, A., Meyunivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. IEEE Trans. Evol. Comput. **6**(2), 182–197 (April 2002)

Deb, K., Goel, T.: A hybrid multi-objective evolutionary approach to engineering shape design. In: Zitzler, E., et al. (eds.) Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science, vol. 1993, pp. 385–399. Springer, New York (2001)

Ehrgott, M., Gandibleux, X.: A survey and annotated bibliography of multi-objective combinatorial optimization. OR Spektrum **22**, 425–460 (2000)

El-Sherbeny, N.: Resolution of a vehicle routing problem with multi-objective simulated annealing method. Ph.D. thesis, Faculté Polytechnique de Mons (2001)

Geiger, M.J.: Genetic algorithms for multiple objective routing. In: MIC'2001, pp. 349–353 (2001)

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

Hansen, M.P.: Tabu search for multiobjective optimization: MOTS. In: Multiple Criteria Decision Making (1997)

Hong, S.-C., Park, Y.-B.: A heuristic for a bi-objective vehicle routing with time window constraints. Int. J. Prod. Econ. **62**, 249–258 (1999)

Ishibuchi, H., Yoshida, T., Murata, T.: Balance between genetic search and local search in hybrid evolutionary multi-criterion optimization algorithms. In: Cantu-Paz, E., et al. (eds.) Genetic and Evolutionary Computation Conference, pp. 1301–1308. Morgan Kaufmann (July 2002)

Jaszkiewicz, A.: Genetic local search for multi-objective combinatorial optimization. Eur. J. Oper. Res. **137**, 50–71 (2002)

Jozefowiez, N., Semet, F., Talbi, E.-G.: Parallel and hybrid models for multi-objective optimization: application to the vehicle routing problem. In: Merelo Guervos, J.J., et al. (eds.) PPSN VII. Lecture Notes in Computer Science, vol. 2439, pp. 271–280. Springer, New York (September 2002)

Knowles, J., Corne, D.: On metrics for comparing nondominated sets. In: Congress on Evolutionary Computation (CEC'2002), vol. 1, pp. 711–726. IEEE Center (2002)

Knowles, J.D.: Local-search and hybrid evolutionary algorithms for Pareto optimization. Ph.D. thesis, University of Reading, Reading, UK (January 2002)

Lee, T., Ueng, J.: A study of vehicle routing problems with load-balancing. Int. J. Phys. Distrib. Logist. Manag. **29**(10), 646–658 (1999)

Lenstra, J.K., Rinnooy Kan, A.H.G.: Complexity of vehicle routing and scheduling problem. Networks **11**, 221–227 (1981)

Lin, S.: Computer solutions of the traveling salesman problem. Bell Syst. Tech. J. **44**, 2245–2269 (1965)

Morse, J.N.: Reducing the size of non-dominated set: pruning by clustering. Comput. Oper. Res. **7**(1–2), 55–66 (1980)

Or, I.: Traveling salesman-type combinatorial problems and their relation to the logistics of blood banking. Ph.D. thesis, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL (1976)

Potvin, J.-Y., Bengio, S.: The vehicle routing problem with time windows part II: genetic search. INFORMS J. Comput. **8**(2), 165–172 (1996)

Prins, C.: A simple and effective evolutionary algorithm for the vehicle routing problem. Comput. Oper. Res. **33**(12), 1985–2002 (2006)

Rahoual, M., Kitoun, B., Mabed, M., Bachelet, V., Benameur, F.: Multicriteria genetic algorithms for the vehicle routing problem with time windows. In: MIC'2001, pp. 527–532 (2001)

Ribeiro, R., Loureno, H.R.: A multi-objective model for a multi-period distribution management problem. In: MIC'2001—4th Metaheuristics International Conference, pp. 97–102 (July 2001)

Rochat, Y., Taillard, E.: Probalistic diversification and intensification in local search for vehicle routing. J. Heuristics **1**, 147–167 (1995)

Serafini, P.: Simulated annealing for multiple objective optimization problems. In: The Tenth International Conference on Multiple Criteria Decision Making, vol. 1, pp. 87–96 (1992)

Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, London (1986)

Sessomboon, W., Watanabe, K., Irohara, T., Yoshimoto, K.: A study on multi-objective vehicle routing problem considering customer satisfaction with due-time (the creation of Pareto optimal solutions by hybrid genetic algorithm). Trans. Jpn. Soc. Mech. Eng. **64**, 1108–1115 (1998)

Toth, P., Vigo, D. (eds.): The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications, vol. 9. SIAM (December 2001)

Ulungu, E.L.: Optimisation combinatoire multicritère: détermination de l'ensemble des solutions efficaces et méthodes interactives. Ph.D. thesis. Université de Mons-Hainaut (1993)

Zitzler, E.: Evolutionary algorithm for multiobjective optimization: methods and applications. Ph.D. thesis. Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (November 1999)

Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (2001)