Adaptive Memetic Algorithm for the Vehicle Routing Problem with Time Windows

Jakub Nalepa
Institute of Informatics
Silesian University of Technology
Akademicka 16
44-100 Gliwice, Poland
jakub.nalepa@polsl.pl

ABSTRACT

This paper presents an adaptive memetic algorithm (AMA) to minimize the total travel distance in the NP-hard vehicle routing problem with time windows (VRPTW). Although memetic algorithms (MAs) have been proven to be very efficient in solving the VRPTW, their main drawback is an unclear tuning of their numerous parameters. Here, we introduce the AMA in which the selection scheme and the population size are adjusted during the search. We propose a new adaptive selection scheme to balance the exploration and exploitation of the search space. An extensive experimental study confirms that the AMA outperforms a standard MA in terms of the convergence capabilities.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

Keywords

Memetic algorithm; self-adaptation; vehicle routing problem with time windows

1. INTRODUCTION

The vehicle routing problem with time windows (VRPTW) is a well-known NP-hard discrete optimization problem. Its applications are of wide range, thus, a number of exact [3] and approximate algorithms to solve the VRPTW emerged over the years. The latter include various improvement and construction heuristics [2], ant colony algorithms, simulated annealing, genetic and memetic algorithms, and more [5].

The VRPTW is defined on the graph G, where each customer v_i, $i \in \{1, 2, \ldots, C\}$ (and the depot v_0) is given as a vertex, and each edge is a travel connection with the cost $c_{i,j}$, $i, j \in \{0, 1, \ldots, C\}$, $i \neq j$. Customers define the demands d_i, $d_i \geq 0$, and service times s_i. The time windows $[e_i, l_i]$ are given for each v_i and the depot. Let K be a number of vehicles with a constant capacity Q in a solution σ. Then, σ is feasible if (i) Q is never exceeded, (ii) each v_i is served within $[e_i, l_i]$ in exactly one route, (iii) every vehicle starts and finishes at v_0 within $[e_0, l_0]$. The primary objective is to minimize K. Also, the total travel distance $T = \sum_{i=1}^{K} T_i$ is to be minimized, where T_i is the distance of the *i-th* route. σ_A is of a higher quality than σ_B if $(K(\sigma_A) < K(\sigma_B))$ or $(K(\sigma_A) = K(\sigma_B)$ and $T(\sigma_A) < T(\sigma_B))$.

The paper is organized as follows. The adaptive memetic algorithm is described in Section 2. The experimental study is reported in Section 3. Section 4 concludes the paper.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). Copyright is held by the author/owner(s).
GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
ACM 978-1-4503-2881-4/14/07.
http://dx.doi.org/10.1145/2598394.2602273.

2. ADAPTIVE MEMETIC ALGORITHM

In the AMA, which extends our previous efforts [1, 6], a population of solutions evolves in time to decrease T. First, K is minimized by the guided ejection search [1], and a population of N_I feasible solutions (each containing K routes) is generated (Alg. 1, line 1). Then, according to a pre-selection scheme $\mathcal{S}$, N pairs (σ_A, σ_B) are determined (line 4) and crossed-over to generate N_c children σ_c for each pair using the edge assembly crossover (EAX) (line 8). If σ_c is infeasible then it is repaired, and it undergoes the education and mutation procedures based on applying local search moves (line 9). The best child σ_c^b is determined for each (σ_A, σ_B) (line 10). Finally, the next population is formed (line 13).

The AMA parameters are adjusted during the search. First, we propose to incrementally increase the population size N, starting from the initial size N_I (with step ΔN). Also, we combine the AB-selection scheme, which proved to have high exploration capabilities [4, 6], with the scheme locally exploiting best individuals (termed *Local Exploitation Selection*, LES). In LES, the population is sorted and divided into ϵ parts. Then, N/ϵ pairs of parents are drawn and crossed-over for each part. The children form the next population of size N. Here, the elitist strategy is applied.

In the AMA, s indicates the number of consecutive generations for which the best solution σ^B was not improved. $\mathcal{S}$ is switched to LES for better local exploitation once s exceeds S, i.e., the maximum steady state selection counter (line 19). If s surpasses P (the maximum steady state population counter), then $\mathcal{S}$ is set back to the AB-selection, and N increases to explore new regions of the search space (line 21). Finally, the best solution is returned (line 26).

3. EXPERIMENTAL RESULTS

The AMA was implemented in C++ and run on an Intel i7 2.3 GHz (16 GB RAM) computer. It was tested on the

Algorithm 1 Adaptive memetic algorithm (AMA).

Generate a population of N_I solutions with K routes;
$done \leftarrow$ false; $N \leftarrow N_I$; $s \leftarrow 0$; $T(\sigma_p^B) \leftarrow \infty$; $\mathcal{S} \leftarrow AB$;
while not *done* **do**
  Determine N pairs (σ_A, σ_B);
  for all (σ_A, σ_B) **do**
    $\sigma_c^b \leftarrow \sigma_A$;
    for i $\leftarrow 1$ **to** N_c **do**
      $\sigma_c \leftarrow$ CrossoverAndRepair(σ_A, σ_B);
      $\sigma_c \leftarrow$ Educate(σ_c); $\sigma_c \leftarrow$ Mutate(σ_c);
      $\sigma_c^b \leftarrow$ UpdateBestChild(σ_c^b, σ_c);
    end for
  end for
  Form the next population of size N and update σ^B;
  if $T(\sigma^B) < T(\sigma_p^B)$ **then**
    $s \leftarrow 0$;
  else
    $s \leftarrow s + 1$;
    if $(s > S$ **and** $s < P)$ **then**
      $\mathcal{S} \leftarrow$ LES;
    else if $s > P$ **then**
      $\mathcal{S} \leftarrow$ AB; $N \leftarrow N + \Delta N$; $s \leftarrow 0$;
    end if
  end if
  $\sigma_p^B \leftarrow \sigma^B$; *done* $\leftarrow$ CheckStoppingCondition();
end while
return best solution;

Gehring and Homberger's (GH) benchmark tests with 200 customers. GH tests are divided into six subclasses: C1, C2 (clustered customers), R1, R2 (random ones), RC1 and RC2 (both random and clustered). The subclasses C1, R1 and RC1 have smaller Q and shorter time windows than C2, R2 and RC2. There are 10 problem instances in each subclass.

Each test (out of 10) in each subclass was executed 5 times, and the best results (i.e., with the minimum T) were averaged for each subclass. The AMA parameters were set as follows: $N_I = \Delta N = 10$, $N_c = 20$, $\epsilon = 5$, $S = 20$, $P = S + 5$. For the MA we used the AB-Selection and $N = 20000/C = 100$, as suggested in [5]. The maximum execution time of the AMA was limited to $\tau = 5$ min.

The experimental results are given in Tab. 1. We compare the best travel distances obtained using the MA, the AMA without the selection scheme adaptation ($S = P$) and the AMA, in various time points τ_i (i stands for minutes). Also, we show the world's best known T averages (WB)[1]. In this study we obtained the world's best known K for each test.

It is easy to note that the initial best solutions (in τ_0) were of the highest quality in the MA, due to its large N – the probability of obtaining a well-fitted individual in the initial population was large. However, the MA, in which N is constant during the search, required much more time to converge to the better solutions compared with both versions of the AMA. They outperformed the MA in $\tau < \tau_1$. Here, the small populations were intensively exploited with LES and extended to include new individuals if necessary. The average populaton sizes N_1 and N_5 (in τ_1 and τ_5) prove that balancing the exploitation and exploration of the search space can be achieved by a smooth growth in N.

[1]See SINTEF website (ref.: March 18, 2014): `http://www.sintef.no/Projectweb/TOP/VRPTW/Homberger-benchmark`.

Table 1: The best results averaged for each subclass.

			C1	C2	R1	R2	RC1	RC2
MA	τ_0	T_0	**2853.57**	**1856.97**	**3988.23**	**3068.85**	**3790.70**	**2686.44**
	τ_1	T_1	2784.95	1848.77	3876.20	3043.86	3607.92	2657.72
	τ_5	T_5	2720.07	1837.44	3721.14	2969.83	3432.39	2575.97
AMA (S=P)	τ_0	T_0	2902.41	1870.89	4063.52	3102.48	3812.95	2705.66
	τ_1	T_1	**2720.06**	1834.78	3672.62	**2952.20**	3344.80	2566.55
		N_1	24	14	12	10	11	10
	τ_5	T_5	2719.64	1832.59	3649.20	2937.43	3272.49	2546.71
		N_5	60	34	30	18	29	19
AMA	τ_0	T_0	2880.22	1857.79	4043.66	3106.33	3842.79	2702.22
	τ_1	T_1	2720.23	**1834.37**	**3671.78**	2957.36	**3334.32**	**2566.31**
		N_1	15	11	13	10	11	10
	τ_5	T_5	**2718.87**	**1831.95**	**3647.68**	**2936.61**	**3270.71**	**2545.80**
		N_5	48	29	29	14	28	17
WB			2718.41	1831.59	3611.86	2929.41	3176.23	2535.88

4. CONCLUSIONS AND FUTURE WORK

We proposed an adaptive memetic algorithm for solving the VRPTW. The population size and the selection scheme are adjusted dynamically during the search in order to balance the exploitation and exploration of the search space. The experimental results proved its high convergence capabilities compared with the standard MA. Our ongoing research includes enhancing the AMA further, and incorporating it into our parallel algorithm.

5. ACKNOWLEDGMENTS

We thank the following computing centers where the computations were carried out: Academic Computer Centre in Gdańsk TASK, Academic Computer Centre CYFRONET AGH, Kraków, Interdisciplinary Centre for Mathematical and Computational Modeling, Warsaw University, Wrocław Centre for Networking and Supercomputing. This work was supported by BKM/515/RAu2/2013 grant from the Institute of Informatics, Silesian University of Technology.

6. REFERENCES

[1] M. Blocho, Z. J. Czech. A parallel memetic algorithm for the vehicle routing problem with time windows. Proc. IEEE 3PGCIC, 2013, pp. 144–151.

[2] O. Bräysy and M. Gendreau. Vehicle Routing Problem with Time Windows: Route Construction and Local Search Algorithms. Trans. Sc. **39**(1) (2005) 104–118.

[3] B. Kallehauge. Formulations and exact algorithms for the vehicle routing problem with time windows. Comput. Oper. Res. **35**(7) (2008) 2307–2330.

[4] M. Kawulok and J. Nalepa. Support vector machines training data selection using a genetic algorithm, Proc. S+SSPR 2012, ser. LNCS. Springer, 2012, vol. 7626, pp. 557–565.

[5] Y. Nagata, O. Bräysy, W. Dullaert. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. Comput. Oper. Res. **37**(4) (2010) 724–737.

[6] J. Nalepa and Z. J. Czech. New selection schemes in a memetic algorithm for the vehicle routing problem with time windows. Proc. ICANNGA 2013, ser. LNCS, Springer, 2013, vol. 7824, pp. 396–405.