Temporally Adaptive Co-operation Schemes

Jakub Nalepa and Miroslaw Blocho

Abstract Selecting an appropriate co-operation scheme in parallel evolutionary algorithms is an important task and it should be undertaken with care. In this paper, we introduce the temporally adaptive schemes, and apply them in our parallel memetic algorithm for solving the vehicle routing problem with time windows. The experimental results revealed that this approach allows for retrieving better solutions in much shorter time compared with other co-operation schemes. The analysis is backed up with the statistical tests, which gave the clear evidence that the results are important. We report one new world's best solution to the benchmark problem obtained using our adaptive co-operation scheme.

Key words: Parallel algorithm; co-operation; memetic algorithm; VRPTW

1 Introduction

Solving rich vehicle routing problems (VRPs) is a vital research topic due to their practical applications which include delivery of food, beverages and parcels, bus routing, delivery of cash to ATM terminals, waste collection, and many others. There exist a plethora of variants of rich VRPs reflecting a wide range of real-life scheduling scenarios [6, 19]—they usually combine multiple realistic constraints which are imposed on feasible solutions. Although exact algorithms retrieve the optimum routing schedules, they are

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

Miroslaw Blocho
Institute of Informatics, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland e-mail: blochom@gmail.com

F. Xhafa et al. (eds.), *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*, Lecture Notes on Data Engineering and Communications Technologies 1, DOI 10.1007/978-3-319-49109-7_14

still very difficult to exploit in practice, because of their unacceptable execution times for massively-large problems. Therefore, approximate algorithms became the main stream of research and development—these approaches aim at delivering high-quality (however not necessarily optimum) schedules in significantly shorter time. In our recent work [14], we showed that our parallel memetic algorithm (PMA–VRPTW)—a hybrid of a genetic algorithm and some local refinement procedures—elaborates very high-quality schedules for the vehicle routing problem with time windows (VRPTW). Although PMA–VRPTW was very efficient, selecting the appropriate co-operation scheme (defining the co-operation topology, frequency and strategies to handle emigrants/immigrants) is extremely challenging and time-consuming—the improper selection can easily jeopardize the PMA–VRPTW capabilities.

1.1 Contribution

We propose two temporally adaptive co-operation schemes in PMA–VRPTW. In these schemes, the master process samples several time points during the execution, and monitors the search progress. Based on this analysis, the scheme is dynamically updated to balance the exploration and exploitation of the solution space, and to guide the search process as best as possible.

Our experiments performed on the well-known Gehring and Homberger's benchmark (in this work, we consider all 400-customer tests with wide time windows, large truck capacities, and random positions of the customers, which appeared very challenging [14]), revealed that the new temporally adaptive co-operation schemes allow for retrieving better solutions quickly (the differences are statistically important), compared with other means of co-operations. We report one new world's best solution elaborated using the new scheme. It is worth mentioning that such temporally adaptive strategies of establishing the desired co-operation schemes have not been intensively studied in the literature so far, and they may become an immediate answer to the problems which require the parallel processes to co-operate efficiently to guide the search process towards high-quality solutions quickly.

1.2 Paper Structure

This paper is structured as follows. Section 2 describes the VRPTW. In Section 3, we review the state of the art on the VRPTW. PMA–VRPTW is briefly discussed in Section 4. In the same section, we present the temporally adaptive co-operation schemes, which are the main contribution of this work. Section 5 contains the analysis of the experimental results. Section 6 concludes the paper and serves as the outlook to the future work.

2 Problem Formulation

The VRPTW is an NP-hard optimization problem of delivering goods to C customers using K homogeneous trucks. The main objective is to minimize the fleet size, and the secondary one is to optimize the total travel distance.

The VRPTW is defined on a complete graph $G = (V, E)$ with vertices V=$\{v_0$,v_1,...,$v_C\}$ (representing the travel points), and edges E=$\{(v_i, v_j)$: $v_i, v_j \in V, i \neq j\}$ (travel connections). The node v_0 is the depot (there is only one depot, i.e., the start and the finish travel point of all trucks). Each v_i defines its non-negative demand q_i (there is no depot demand, thus $q_0 = 0$), service time s_i ($s_0 = 0$), and time window $[e_i, l_i]$ (the service must be *started* within this slot, however it may finish after the time window has been closed). Every edge (v_i, v_j) has a travel cost c_{ij} (given in the Euclidean metric). A feasible solution is a set of K routes such that: (i) each route starts and ends at the depot, (ii) the truck loads do not exceed Q, (iii) the service of each v_i begins between e_i and l_i, (iv) each truck returns to the depot before l_0, and (v) each customer is served in exactly one route. If any of the constraints is violated, then the solution becomes unacceptable.

Let (K_α, T_α) and (K_β, T_β) represent two feasible VRPTW solution, denoted as α and β, respectively. The solution β is of a higher quality than the solution α, if $(K_\beta < K_\alpha)$ or $(K_\beta = K_\alpha$ and $T_\beta < T_\alpha)$. Hence, the solution β encompasses a lower number of routes, or—if the numbers of trucks are equal for both α and β—the total distance traveled during the service is smaller.

An exemplary solution σ of the VRPTW instance containing 25 customers is visualized in Fig. 1. This solution consists of three routes (r_1, r_2, and r_3): $r_1 = \langle v_0, v_8, v_{10}, v_{21}, v_{12}, v_{22}, v_{23}, v_{24}, v_{25}, v_{17}, v_{14}, v_0\rangle$ (10 customers are visited), $r_2 = \langle v_0, v_{11}, v_{15}, v_{19}, v_{20}, v_{18}, v_{16}, v_9, v_{13}, v_7, v_0\rangle$ (9 customers), and $r_3 = \langle v_0, v_6, v_2, v_1, v_4, v_3, v_5, v_0\rangle$ (6 customers). It is easy to see that each customer v_i, $i \in \{1, \ldots, 25\}$, is served exactly once (i.e., in one route). Assuming that the vehicle loads do not exceed the capacity in any route, and the time window constraints are not violated, this routing schedule is feasible.

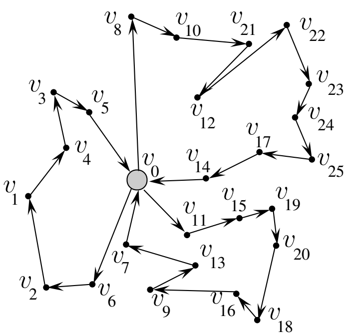

Fig. 1 An exemplary solution to the VRPTW instance with 25 clients served in 3 routes.

3 Related literature

Due to its wide practical applicability, the VRPTW attracted research attention. Exact algorithms aim at delivering the optimum solutions, however they are still difficult to apply in practice, because of their unacceptable execution times. These approaches encompass branch-and-cut, branch-and-bound, dynamic programming solutions, along with a plethora of various VRPTW formulations [1]. Exact algorithms were summarized and thoroughly discussed in numerous interesting surveys and reviews [2,8]. It is worth mentioning that in a majority of such approaches, minimizing the total distance is considered as the single objective.

The approximate methods include construction (creating solutions from scratch [20]) and improvement (which boost the quality of initial, usually very low-quality solutions [5, 11]) heuristics, and various meta-heuristics (very often allowing for the temporary deterioration of the solution quality during the optimization process) [5], including ant colony optimization techniques [7], particle swarm-based approaches [9], neighborhood searches [10], and many others [3]. In genetic algorithms (GAs), a population of solutions (chromosomes) undergoes the evolution in search of well-fitted individuals representing high-quality feasible solutions [21].

Memetic algorithms (MAs) combine EAs for exploring the entire search space, with intensive refinement procedures applied to exploit solutions already found [17] (they are often referred to as hybrid GAs). Such approaches have been successfully applied for solving a wide spectrum of optimization and pattern recognition problems [23]. A number of sequential and parallel MAs have been proposed for tackling the VRPTW [12,16,22], as well as other challenging rich VRPs [13, 18].

In our recent work [14], we showed that the co-operation scheme has a tremendous impact on the quality of final VRPTW solutions, and on the convergence time in our co-operative parallel MA. Its appropriate selection is not trivial and should respond to the search state. Also, we showed that dividing the search space across the co-operating processes (referred to as *islands*) helps significantly improve the exploration capabilities of the parallel algorithm [4, 15]. In this work, we tackle the problem of retrieving the appropriate co-operation schemes on the fly. This should allow for responding to the current search progress, and for choosing the best-fitted co-operation scheme (either explorative or exploitative). Such approaches have not been intensively explored in the literature so far.

4 Parallel Algorithm

In PMA–VRPTW (Algorithm 1)—which is a homogeneous island model parallel MA, since each island (a parallel process) runs the same MA to minimize

T—each individual p_i, where $i \in \{1, 2, \ldots, N\}$, corresponds to a VRPTW solution with K routes in a population of N solutions (on each island). The initial populations are generated using the parallel guided search [14] (it minimizes K at first, and then is used to create initial populations for each island). These populations evolve to optimize the distance T (lines 2–16).

Algorithm 1 Parallel memetic algorithm (PMA–VRPTW).

```
 1: Minimize K and find populations for each island;
 2: parfor P_i ← P_1 to P_n do
 3:     while not finished do
 4:         Determine N pairs (p_a, p_b);
 5:         for all (p_a, p_b) do
 6:             GenerateChild(p_a, p_b);                    ▷ Fig. 2
 7:         end for
 8:         Form the next population of size N;
 9:         if (can co-operate) then
10:             Determine and send emigrant(s);
11:             Receive and handle immigrant(s);
12:         end if
13:         Verify termination condition;
14:     end while
15: end parfor
16: return best solution among all islands;
```

The evolution involves selecting pairs of individuals for crossover, recombining them using the edge-assembly operator [12], and restoring the feasibility of children if it is necessary, using local edge-exchange moves (Fig. 2). Then, the children are *educated* (this is a memetic operator, thus it is rendered in light red), and mutated. Both operations involve applying edge-exchange and edge-relocate moves. The islands co-operate (Algorithm 1, lines 9–12), to propagate the best solutions found up to date, and to guide the search towards better routing schedules. The best individual (across all processes) is finally returned (line 16). For more details on PMA–VRPTW, see [14].

4.1 Temporally Adaptive Co-operation

In the temporally adaptive co-operation schemes (which are based upon our previous knowledge synchronization and ring schemes [14]), we monitor the dynamic changes of the total distance T of the best solution in the master island. During each co-operation phase (which occurs after finishing each generation), we calculate the differences:

$$\Delta T_i = \left| G_{(c-i)}(T) - G_c(T) \right|, \tag{1}$$

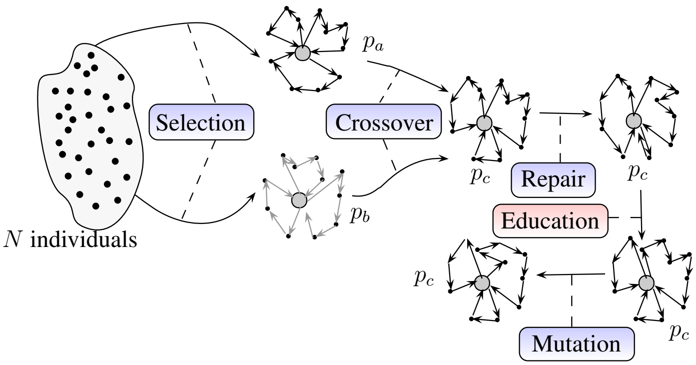

Fig. 2 Creation of a child in PMA–VRPTW.

where $G_{(c-i)}(T)$ denotes the best travel distance in the G_{c-i} generation (let G_c be the current generation). The ΔT_i values are found for three time points in the past—it is visualized in Fig. 3 (the differences are found for the second, fifth, and tenth generation before G_c, shown in blue).

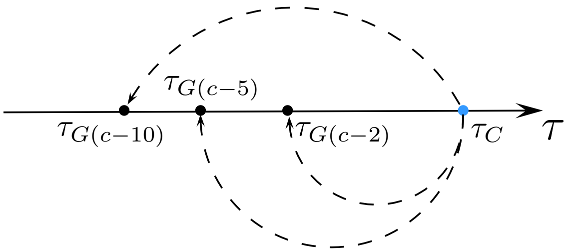

Fig. 3 Sampling several T values during the evolution.

Certain differences are compared with the expected improvements in the travel distances (ΔT_i^e). These comparisons are exploited to adapt the scheme—if the current co-operation is explorative, then it may be appropriate to switch it to the more exploitative one (and vice versa). In both phases (minimizing K and T), the more exploitative version of the scheme (either ring or KS) is used at first (we exploit only 10% of the closest customers to the one being affected in the edge-exchange moves).

In each co-operation phase, we calculate ΔT_2, ΔT_5, and ΔT_{10} (the last increment is found only in the exploitation mode, whereas the first—in the exploration), along with the expected improvements. For the exploitative co-operations, we have: ΔT_5^e and ΔT_{10}^e, where $\Delta T_5^e = \alpha_5 G_{(c-5)}(T)$, and $\Delta T_{10}^e = \alpha_{10} G_{(c-10)}(T)$, and the α coefficients are given in %, whereas for the explorative ones (i.e., ring or KS with the search space partitioning [15]) we additionally have the lower bounds of these measures ($\beta_2 \Delta T_2^e$ and $\beta_5 \Delta T_5^e$, where β's are in %). These expected improvements are thus dependent on the travel distance in the $G_{(c-i)}$ generation, denoted as $G_{(c-i)}(T)$, and on the current co-operation mode (exploration or exploitation). Note that the α's may differ for both co-operation modes.

Algorithm 2 Temporal adaptation of the co-operation.

1: **if** (exploitation mode) **then**
2:     **if** ($\Delta T_2 = 0$ **or** $\Delta T_5 \leq \Delta T_5^e$ **or** $\Delta T_{10} \leq \Delta T_{10}^e$) **then**
3:         Switch to explorative co-operation;
4:     **end if**
5: **else**
6:     **if** ($\Delta T_5 = 0$ **or**
            $\Delta T_5 \geq \Delta T_5^e$ **or** $\Delta T_5 \leq \beta_5 \Delta T_5^e$ **or**
            $\Delta T_2 \geq \Delta T_2^e$ **or** $\Delta T_2 \leq \beta_2 \Delta T_2^e$) **then**
7:         Switch to exploitative co-operation;
8:     **end if**
9: **end if**

Algorithm 2 presents the adaptation process. If the changes in the best T value are relatively small in the exploitation mode, then it is switched to the explorative one (line 3). On the other hand, if these changes are significant during the exploration, it indicates that this part of the solution space should be more intensively exploited, hence the co-operation toggles its mode (line 7). Also, if they are very small (less than the lower bounds), then the further exploration may not help find new high-quality solutions, and the mode becomes exploitative (this often happens when the high-quality solutions have already been retrieved).

5 Experimental Validation

5.1 Settings

PMA–VRPTW was implemented in the C++ programming language using the Message Passing Interface (MPI). The computations were carried out on the cluster equipped with Intel Xeon Quad Core 2.33 GHz processors, each with 12 MB level 3 cache. The nodes were connected by the Infiniband DDR fat-free network (throughput 20 Gbps, delay 5 μs). The source code was compiled using Intel 10.1 compiler and MPICH v. 1.2.6 MPI library.

We compared the proposed temporally adaptive schemes with our previous (best) ones [14] (in Table 1, we gather the co-operation schemes which have been investigated in this work). For the exploitation mode, we have: $\alpha_5 = 0.2\%$, and $\alpha_{10} = 0.5\%$, whereas for the exploration: $\alpha_2 = 1\%$, $\alpha_5 = 2\%$, and $\beta_2 = \beta_5 = 25\%$—the α and β parameter values were tuned experimen-

Table 1 Investigated co-operation schemes.

(a)	Ring
(b)	Ring with partitioned neighborhoods
(c)	Ring with partitioned routes
(d)	Ring with both partitioning strategies
(e)	Knowledge synchronization
(f)	Knowledge synchronization with partitioned neighborhood
(g)	Knowledge synchronization with partitioned routes
(h)	Knowledge synchronization with both partitioning strategies
(i)	**Adaptive knowledge synchronization**
(j)	**Adaptive ring**

tally, using test instances of various characteristics and structures. However, while selecting the appropriate α and β values, it is necessary to analyze the underpinning ideas of the current co-operation scheme (note that the α parameters affect the change from the exploitative to the explorative mode, whereas the β coefficients—from the explorative to the exploitative one). In the exploitative mode, the changes in T's are most often notably smaller compared with those retrieved in the explorative mode. This observation may become a good starting point in the tuning process of these parameters, however it requires further research attention. In all experiments, the number of processes was $n = 24$, the maximum evolution time was set to $\tau_E = 2000$ seconds, and the maximum time of minimizing K was $\tau_K = 60$ seconds (the first phase took approximately 10 seconds in all cases).

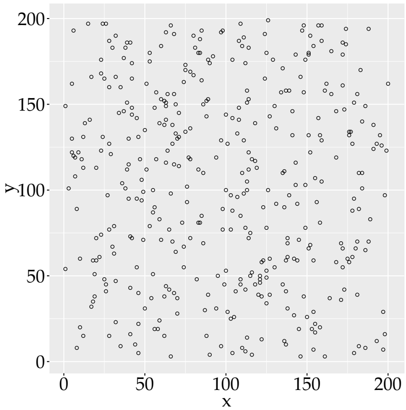

Fig. 4 An exemplary structure of a 400-customer Gehring and Homberger's test instance with the customers randomly scattered around the map.

In this work, we focus on 400-customer Gehring and Homberger's tests with random positions of travel points, wide time windows, and relatively large truck capacities (class r2). An exemplary structure of a test belonging to this class of benchmark instances is visualized in Fig. 4.

5.2 Analysis and Discussion

The results obtained using PMA–VRPTW with various co-operations are gathered in Table 2. We sampled and averaged the best T's (across all islands) in several time points (PMA–VRPTW was executed 10× using each scheme for each—out of 10—problem instance). Our new adaptive schemes significantly outperformed other ones. Importantly, PMA–VRPTW with the new schemes converged to very high-quality schedules quickly—the average T in $\tau = 30$ minutes is reduced by approx. 0.7% and 0.3% compared with $\tau = 5$ minutes for the adaptive KS and ring, respectively. Hence, this decrease is negligible. Therefore, the algorithm could have been terminated much earlier, since acceptable solutions had already been retrieved. It is worth noting that we have beaten the world's best solution (we decreased T from 7129.03 to 7128.93 for $K = 8$) for the r2_4_5 test using the adaptive KS[1].

Table 2 The average travel distances T (the best results out of 10 independent executions of PMA–VRPTW with each co-operation scheme applied are averaged for 10 instances in the r2 class). The best T's (in each sampled time point) are boldfaced.

Scheme	$\tau = 5$ min.	$\tau = 10$ min.	$\tau = 15$ min.	$\tau = 20$ min.	$\tau = 25$ min.	$\tau = 30$ min.
(a)	6265.73	6200.77	6190.64	6189.97	6189.91	6189.91
(b)	6205.87	6195.02	6191.54	6189.57	6189.25	6188.80
(c)	6284.24	6219.63	6199.34	6193.09	6189.83	6187.72
(d)	6199.87	6193.27	6191.77	6191.30	6163.37	6190.35
(e)	6353.40	6257.54	6218.06	6199.36	6191.10	6186.31
(f)	6195.94	6185.86	6183.25	6181.51	6180.35	6179.72
(g)	6329.54	6247.26	6212.84	6197.85	6192.12	6189.56
(h)	6195.58	6180.52	6178.54	6177.81	6177.56	6177.01
(i)	6171.40	6168.65	**6167.72**	**6167.37**	**6167.13**	**6166.89**
(j)	**6169.91**	**6167.77**	**6167.72**	6167.65	6167.62	6167.62

In Fig. 5, we render the average convergence time (i.e., after which the best solution across all co-operating islands could not be further improved, and may be considered as the *target solution*) of the T optimization phase. Applying the temporally adaptive schemes allowed for decreasing this time notably (also, the retrieved solutions were of a much higher quality—see Table 2). In the average case, PMA–VRPTW converges up to 3.7× faster when the adaptive ring scheme is applied, compared with our previous co-operations. It is quite important in practical applications, in which high-quality routing schedules should be retrieved as fast as possible. The results show that converging to target solutions is significantly faster when the adaptation is applied—see e.g., Fig. 5(j) (adaptive ring) compared with Fig. 5(a,d) (ring and ring with both partitioning strategies—almost 1.9× faster on average), with Fig. 5(b) (ring with partitioned neighborhoods—almost 2× faster), and with Fig. 5(c) (ring with partitioned routes—2.5× faster). Similarly, the adaptive KS is up

[1] The details can be found at: `http://sun.aei.polsl.pl/~jnalepa/3PGCIC16`.

to 2.5× faster than KS (on average). Finally, the best and the worst convergence times are also the lowest in the case of adaptive co-operation schemes (see the orange and gray bars in Fig. 5). Since the routing schedules retrieved using these schemes are of the highest-quality (as shown in Table 2), these schemes outperform the other ones when both the convergence time and the quality of final solutions are considered.

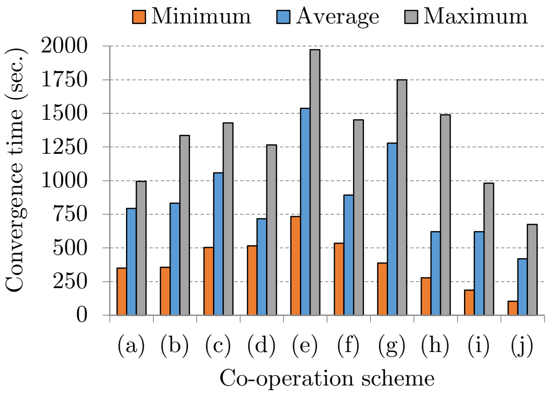

Fig. 5 Convergence time (in seconds) of PMA–VRPTW for various co-operation schemes.

Finally, we performed the two-tailed Wilcoxon tests to verify the null hypothesis saying that "applying different co-operation schemes leads to retrieving solutions of the same quality". The levels of the statistical significance are presented in Table 3—they prove that using our new temporally adaptive schemes allows for elaborating significantly different (better) routing schedules (the null hypothesis can be safely rejected because $p < 0.0001$ in most cases). Although the differences between the schedules obtained using two adaptive schemes (adaptive ring and adaptive KS) are not necessarily statistically important, the adaptive ring should be preferred since it converges faster compared with the adaptive KS (see Fig. 5).

6 Conclusions and Outlook

In this paper, we proposed two temporally adaptive co-operation schemes, and applied them in our parallel algorithm for solving the VRPTW. The adaptation procedure involves monitoring of the search process, and the dynamic selection of the appropriate co-operation strategy—this strategy may exhibit either more explorative or more exploitative behavior, depending on the current optimization state. The experimental study performed on the Gehring and Homberger's benchmark tests with randomized customers re-

Table 3 The level of statistical significance obtained using the two-tailed Wilcoxon tests. The differences which are statistically important (at $p < 0.05$) are boldfaced.

	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
(a)	0.0949	**0.0061**	0.3173	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**
(b)	—	**0.0019**	0.1556	**0.0003**	**<0.0001**	**0.001**	**0.0001**	**<0.0001**	**<0.0001**
(c)		—	**0.0164**	**0.0001**	**<0.0001**	**0.0016**	**<0.0001**	**<0.0001**	**<0.0001**
(d)			—	**0.0009**	**<0.0001**	**0.0008**	**<0.0001**	**<0.0001**	**<0.0001**
(e)				—	**<0.0001**	**0.0375**	**<0.0001**	**<0.0001**	**<0.0001**
(f)					—	**<0.0001**	0.0767	**<0.0001**	**<0.0001**
(g)						—	**<0.0001**	**<0.0001**	**<0.0001**
(h)							—	**<0.0001**	**<0.0001**
(i)								—	0.332

vealed that utilizing the proposed schemes allows for retrieving solutions of a higher quality (the differences are statistically important) in much shorter time. We reported one new world's best solution elaborated using PMA–VRPTW with the new temporally adaptive scheme.

Our future work is focused on applying our new adaptive schemes for solving other challenging optimization problems (especially, the pickup and delivery with time windows). The presented ideas are quite generic and could be applied in parallel algorithms for other tasks too. Also, we plan to complement the suggested co-operation schemes with the adaptation of the co-operation frequency (similarly, based on the temporal analysis of the search progress). We work on the automatic selection of the most appropriate points to sample the T values in the adaptive schemes, as well as on the adaptation of their parameters. We plan to perform the full scalability tests using the large-scale parallel systems (e.g., computational clusters). Finally, it will be interesting to investigate how the co-operation schemes affect the diversity of the populations (of all islands) during the PMA–VRPTW execution.

7 Acknowledgments

This research was supported by the National Science Centre under research Grant No. DEC-2013/09/N/ST6/03461, and performed using the Intel CPU and Xeon Phi platforms provided by the MICLAB project No. POIG.02.03.00.24-093/13. We also thank the Gdańsk Computer Centre (TASK CI), where the computations were carried out.

References

1. R. Baldacci, A. Mingozzi, and R. Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5):1269–1283, 2011.

2. R. Baldacci, A. Mingozzi, and R. Roberti. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European J. of Op. Research*, 218(1):1 – 6, 2012.
3. R. Banos, J. Ortega, C. Gil, A. L. Márquez, and F. de Toro. A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows. *Computers & Industrial Engineering*, 65(2):286 – 296, 2013.
4. M. Blocho and J. Nalepa. A parallel algorithm for minimizing the fleet size in the pickup and delivery problem with time windows. In *Proc. EuroMPI*, pages 15:1–15:2, New York, USA, 2015. ACM.
5. O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part II: Metaheuristics. *Transportation Science*, 39(1):119–139, 2005.
6. J. Caceres-Cruz, P. Arias, D. Guimarans, D. Riera, and A. A. Juan. Rich vehicle routing problem: Survey. *ACM Computing Surveys*, 47(2):32:1–32:28, 2014.
7. D. Coltorti and A. E. Rizzoli. Ant colony optimization for real-world vehicle routing problems. *SIGEVOlution*, 2(2):2–9, 2007.
8. N. A. El-Sherbeny. Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *J. of King Saud University*, 22(3):123 – 131, 2010.
9. W. Hu, H. Liang, C. Peng, B. Du, and Q. Hu. A hybrid chaos-particle swarm optimization algorithm for the vehicle routing problem with time window. *Entropy*, 15(4):1247–1270, 2013.
10. B. Jarboui, A. Sifaleras, A. Rebai, M. Bruglieri, F. Pezzella, O. Pisacane, and S. Suraci. A variable neighborhood search branching for the electric vehicle routing problem with time windows. *Electronic Notes in Discrete Mathematics*, 47:221 – 228, 2015.
11. Y. Nagata and O. Bräysy. A powerful route minimization heuristic for the vehicle routing problem with time windows. *Operations Res. Letters*, 37(5):333 – 338, 2009.
12. Y. Nagata, O. Bräysy, and W. Dullaert. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, 37(4):724 – 737, 2010.
13. Y. Nagata and S. Kobayashi. *A Memetic Algorithm for Pickup and Delivery with Time Windows Using Selective Exchange Crossover*, pages 536–545. Springer, 2010.
14. J. Nalepa and M. Blocho. Co-operation in the parallel memetic algorithm. *International Journal of Parallel Programming*, 43(5):812–839, 2015.
15. J. Nalepa and M. Blocho. A parallel algorithm with the search space partition for the pickup and delivery with time windows. In *Proc. 3PGCIC*, pages 92–99, 2015.
16. J. Nalepa and M. Blocho. Adaptive memetic algorithm for minimizing distance in the vehicle routing problem with time windows. *Soft Computing*, 20(6):2309–2327, 2016.
17. J. Nalepa and M. Kawulok. Adaptive memetic algorithm enhanced with data geometry analysis to select training data for SVMs. *Neurocomputing*, 185:113 – 132, 2016.
18. S. U. Ngueveu, C. Prins, and R. W. Calvo. An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 37(11):1877 – 1885, 2010. Metaheuristics for Logistics and Vehicle Routing.
19. E. Osaba, X.-S. Yang, F. Diaz, E. Onieva, A. D. Masegosa, and A. Perallos. A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy. *Soft Computing*, pages 1–14, 2016.
20. K.-W. Pang. An adaptive parallel route construction heuristic for the vehicle routing problem with time windows. *Exp. Sys. with App.*, 38(9):11939 – 11946, 2011.
21. P. Repoussis, C. Tarantilis, and G. Ioannou. Arc-guided evolutionary algorithm for vehicle routing problem with time windows. *Evol. Comp., IEEE Trans. on*, 13(3):624–647, 2009.
22. T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, 40(1):475 – 489, 2013.
23. S. Wrona and M. Pawelczyk. Controllability-oriented placement of actuators for active noise-vibration control of rectangular plates using a memetic algorithm. *Archives of Acoustics*, 38(4):529–536, 2013.