LA-UR- 10-02308

*Approved for public release; distribution is unlimited.*

| | |
|---|---|
| *Title:* | Spatial and Temporal Decompositions For Large-Scale Vehicle Routing with Time Windows |
| *Author(s):* | Russell Bent<br>Pascal Van Hentenryck |
| *Intended for:* | Submission to the 16th International Conference on Principles and Practices of Constraint Programming |

Los Alamos
NATIONAL LABORATORY
EST. 1943

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Form 836 (7/06)

# Spatial, Temporal, and Hybrid Decompositions For Large-Scale Vehicle Routing with Time Windows

Russell Bent[1] and Pascal Van Hentenryck[2]

[1] Los Alamos National Laboratories
[2] Brown University

**Abstract.** This paper studies the use of decomposition techniques to quickly find high-quality solutions to large-scale vehicle routing problems with time windows. It considers an adaptive decomposition scheme which iteratively decouples a routing problem based on the current solution. Earlier work considered vehicle-based decompositions that partitions the vehicles across the subproblems. The subproblems can then be optimized independently and merged easily. This paper argues that vehicle-based decompositions, although very effective on various problem classes also have limitations. In particular, they do not accommodate temporal decompositions and may produce spatial decompositions that are not focused enough. This paper then proposes customer-based decompositions which generalize vehicle-based decouplings and allows for focused spatial and temporal decompositions. Experimental results on class R2 of the extended Solomon benchmarks demonstrates the benefits of the customer-based adaptive decomposition scheme and its spatial, temporal, and hybrid instantiations. In particular, they show that customer-based decompositions bring significant benefits over large neighborhood search in contrast to vehicle-based decompositions.

## 1 Introduction

The scale of optimization problems and the need for finding high-quality solutions has grown steadily in recent years as optimization systems are increasingly deployed in operational, integrated settings. This trend generates significant issues for optimization research, changing the focus from finding optimal solutions to delivering high-quality solutions under time constraints. This paper examines the underlying algorithmic issues in the context of multiple vehicle routing with time windows (VRPTWs), which arise in many transportation applications including courier services, the scheduling of repairs in telecommunication companies, and supply-chain logistics. VRPTWs are particularly interesting in this respect, since instances with as few as 100 customers have not been solved optimally despite intense research [?]. Hence finding high-quality solutions under time constraints for problems with 1,000 customers is a significant challenge.

Spatial and temporal decouplings [17] are natural avenues for speeding up optimization algorithms. Unfortunately, they do not apply easily to large-scale VRPTWs that involve complex spatial and temporal dependencies. To remedy this limitation, the concept of adaptive decoupling was proposed in [4]. Its key idea is to iteratively select subproblems that are optimized independently and reinserted into an existing

solution. The successive decouplings are adaptive as they depend on the current solution, not simply the instance data. The benefits of this approach were demonstrated by a vehicle-based adaptive spatial decomposition VASD scheme which produces high-quality solutions significantly faster than large neighborhood search (LNS) on the class RC1 of the extended Solomon benchmarks. Informally speaking, the VASD scheme partitions the vehicles of an existing solution to obtain two subproblems, reoptimizes one of these subproblems using say LNS, and reinsert the optimized vehicle routes to obtain a new solution. The VASD scheme is attractive since it makes it easy to merge the solutions of decoupled problems. However, it also has a number of limitations. Because it is vehicle-based, it is not as spatially focused as possible since vehicles may often travel across large regions, especially early in the optimization process. Moreover, vehicle-based decompositions cannot really accommodate temporal decouplings, since vehicles generally serve customers with a wide variety of time windows.

This paper remedies these limitations and proposes a customer-based adaptive decomposition (CAD) scheme which can be naturally instantiated to spatial, temporal, and hybrid decouplings. Its key idea is to select a set of customers based on a spatial, temporal, or hybrid property and to define a generalized multi-depot VRPTW involving these customers only. The CAD scheme thus allows for more focused spatial decompositions, tight temporal decompositions, or a combination thereof. The generalized VRPTW is also designed to allow for an easy merging of its reoptimized solution into the existing solution.

The benefits of the CAD scheme are demonstrated on the class R2 of the extended Solomon benchmarks. The experimental results indicate that the CAD scheme significantly outperforms LNS and the VASD scheme on this class. They also indicate the complementarity between spatial and temporal decompositions and hence the value of hybrid decompositions.

The rest of this paper is organized as follows. It first reviews VRPTWs and the adaptive decomposition scheme. It then presents the earlier work on vehicle-based adaptive spatial decompositions and the novel contributions on customer-based adaptive decouplings. The paper then presents several instantiations of the CAD scheme, including spatial, temporal, and randomized decouplings. The experimental results and the related work concludes the paper.

## 2 VRPTWs

A VRPTW instance is specified by a set $\mathcal{C}$ of customers, a set of departure depots $\mathcal{D}^-$, a set of arrival depots $\mathcal{D}^+$, and a set of vehicles $\mathcal{V}$ such that $|\mathcal{D}^-| = |\mathcal{D}^+| = |\mathcal{V}|$. A single depot problem is easily generalized into a multi-depot problem by creating multiple depots at the same location. We use multiple depots since it enables us to specify decoupled problems as VRPTWs. The sites of the VRPTW instance are elements of $Sites = \mathcal{C} \cup \mathcal{D}^- \cup \mathcal{D}^+$. Every site $c$ has a demand $q_c \geq 0$ and a service time $s_c \geq 0$. The travel cost between sites $i$ and $j$ is $t_{ij}$. Each site $c$ has a time window $[e_c, l_c]$ constraining when it can be visited, where $e_c$ and $l_c$ represent the earliest and latest arrival times. Vehicles must arrive at site $c$ before the end of the time window $l_c$. They may ar-

rive early but they have to wait until time $e_c$ to be serviced. Each vehicle has a capacity $Q$.

Solutions are specified in terms of vehicle routes and routing plans. A vehicle route starts from a depot $d^-$, visits a number of customers at most once, and returns to a depot $d^+$. It is thus a sequence $\langle d^-, c_1, \ldots, c_n, d^+ \rangle$ where all sites are different. The customers of a route $r = \langle d^-, c_1, \ldots, c_n, d^+ \rangle$, denoted by *cust(r)*, is the set $\{c_1, \ldots, c_n\}$ and the route $r$ of a customer in $\{c_1, \ldots, c_n\}$ is denoted by $route(c)$. The size of a route, denoted by $|r|$, is $|cust(r)|$. The demand of a route, denoted by $q(r)$, is the sum of the demands of its sites. A route satisfies its capacity constraint if $q(r) \leq Q$. We use $q(c)$ to denote the amount of capacity used by a route up to site $c$. The travel cost *t(r)* of a route $r = \langle d^-, c_1, \ldots, c_n, d^+ \rangle$ is the cost of visiting all its sites.

A routing plan is a set of routes in which every customer is visited exactly once and every depot at most once. Observe that a routing plan assigns a unique earliest arrival time $a_c$ for each site $c$. It also assigns a unique return time $a(r)$ to its destination depot $d^+$ for each route $r$. The routing plan also assigns a departure time for each site $c$, denoted by $\delta_c$. The routing plan also assigns a critical arrival time for each site $c$, denoted by $z_c$. This is the latest time a vehicle can feasibly arrive at $c$.

A solution to the VRPTW is a routing plan $\sigma$ satisfying the capacity and time window constraints, i.e.,

$$\forall r \in \sigma : \; q(r) \leq Q \; \& \; \forall c \in \; Sites : a_c \leq l_c.$$

The ordering of the customers on a route in $\sigma$ implicitly defines a predecessor and successor site for each site $c$, denoted by $pred(\sigma, c)$ and $succ(\sigma, c)$. When the context is clear, $\sigma$ is dropped from the notation for brevity. The size $|\sigma|$ of a routing plan $\sigma$ is the number of non-empty routes in $\sigma$. The VRPTW problem consists of finding a solution $\sigma$ which minimizes a lexicographic function consisting of the number of vehicles and the total travel cost, i.e., $f(\sigma) = \langle |\sigma|, \sum_{r \in \sigma} t(r) \rangle$. Modern algorithms for the VRPTW are often organized in two stages, first minimizing the number of vehicles and then minimizing travel distance [2, 19].

## 3 The Adaptive Decomposition Scheme

This paper aims at finding decouplings to speed up the solving of large-scale VRPTWs. The goal of the decouplings is to decompose a VRPTW $\mathcal{P}$ into two sub-VRPTWs $\mathcal{P}_o$ and $\mathcal{P}_s$ that can be solved independently and whose solutions can be merged into a solution of $\mathcal{P}$. In general, finding static decompositions is difficult. For this reason, we proposed in [4] to use the current solution $\sigma$ of $\mathcal{P}$ to find a decoupling $(\mathcal{P}_o, \mathcal{P}_s)$ with projected solution $\sigma_o$ and $\sigma_s$. The VRPTW $\mathcal{P}_o$ is then reoptimized and its solution is merged with $\sigma_s$ to obtain a new solution to $\mathcal{P}$. More precisely, the Adaptive Decomposition Scheme (ADS) is based on two main principles:

1. Starting from plan $\sigma_0$, it produces a sequence of plans $\sigma_1, \ldots, \sigma_j$ such that $f(\sigma_0) \geq f(\sigma_1) \geq \ldots \geq f(\sigma_j)$.
2. At step $i$, the scheme uses $\sigma_{i-1}$ to obtain a decoupling $(\mathcal{P}_o, \mathcal{P}_s)$ of $\mathcal{P}$ with projected solutions $\sigma_o$ and $\sigma_s$. It reoptimizes $\mathcal{P}_o$ to obtain $\sigma_o^*$ and the new plan $\sigma_i = \text{MERGE}(\sigma_o^*, \sigma_{i-1})$

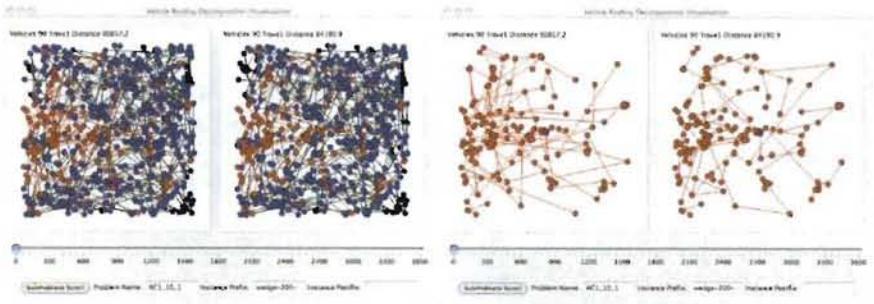


**Fig. 1.** The First Decoupling of VSAD.

One of the most challenging aspects of ADS is how to perform the merging of the decoupled solutions, i.e, $\sigma_i = \text{MERGE}(\sigma_o^*, \sigma_{i-1})$. In [4], we addressed this challenge by choosing $\mathcal{P}_o$ such that the customers of entire vehicles are removed. The merging operation is then trivial, since the vehicles in ($\mathcal{P}_o$ and $\mathcal{P}_s$) are disjoint. We now review this scheme to emphasize its strengths and limitations.

## 4 Vehicle-Based Spatial Adaptive Decompositions

The decomposition presented in [4] is a vehicle-based adaptive decoupling (VAD). It partitions the vehicles to obtain $\mathcal{P}_o$ and $\mathcal{P}_s$, reoptimizes $\mathcal{P}_o$, and uses the new optimized routes, and the routes in $\mathcal{P}_o$ to obtain a new solution. Only spatial decompositions were considered in [4]. The idea was to view the customer region as a circle, randomly selects a wedge $W$, and partitions the vehicles into those serving at least one customers in $W$ and the others. The resulting Vehicle-Based Spatial Adaptive Decomposition VSAD is particularly effective and produced high-quality solutions quickly on instances with up to 1,000 vertices. Its main benefits are the simple definition of $\mathcal{P}_o$ and the trivial implementation of merging, which simply uses the optimized routes of $\mathcal{P}_o$ to replace the old routes in the existing solution.

The VAD scheme has a number of limitations however. First, because the decoupling is vehicle-based, the customers can be located significantly outside the selected wedge. This is illustrated in Figure 1 which depicts the behavior of the VASD scheme visually. The left part of Figure 1 shows the initial plan $\sigma_0$ (left) and the plan $\sigma_1$ (right) after the first decoupling and optimization. The customers in the subproblem $\mathcal{P}_o$ are in red, the remaining ones in blue. The right part of Figure 1 shows the projected solution $\sigma_o$ for subproblem $\mathcal{P}_o$ (bottom left) and its reoptimization $\sigma_o^*$ (bottom right). As can be seen, the first subproblem is quite spread out, illustrating the spatial decomposition is not as tight as desired.

More important however is the fact that the VAD scheme does not scale to other decomposition criteria and, in particular, to temporal decompositions. Indeed, unless the time windows are wide, it is very unlikely that good solutions cluster customers with

similar time windows on the same vehicle, since the vehicle will be inactive for most of the time horizon. Since it uses a vehicle-based decomposition, the VASD scheme is not well-adapted to exploit temporal locality.

## 5 Customer-Based Adaptive Decompositions

To remedy this limitation, this paper proposes a Customer-based Adaptive Decomposition (CAD) scheme. A decoupled problem in the CAD scheme is given by a set of subsequences of customers and has a new set of depots and constraints so that the solutions of $\sigma_o^*$ can be inserted into $\sigma_s$, while ensuring feasibility of the resulting plan.

Given a sequence of customers $\langle c_i, \ldots, c_j \rangle$ for the decoupling, the depots of the subproblem are constructed as follows:

- $d^- = pred(c_i)$: the origin depot is the predecessor of the sequence.
- $d^+ = succ(c_j)$: the destination depot is the successor of the sequence.
- $e_{d^-} = \delta_{pred(c_i)}$: the departure time of $c_i$ is the earliest departure time for $d^-$.
- $l_{d^+} = z_{succ(c_j)}$: the critical arrival time of $succ(c_j)$ is the latest departure for $d^+$.
- $q_{d^-} = q(pred(c_i))$: the demand of $d^-$ is the cumulative demand up to $pred(c_i)$.
- $q_{d^+} = q(succ(c_j)) - q(c_j)$: the demand of $d^+$ is the cumulative demand after $c_j$.

By constructing depots using the border regions of a sequence, any feasible route between $d^-$ and $d^+$ can be reinserted between $pred(c_i)$ and $succ(c_j)$ of $\sigma_{i-1}$, while maintaining the feasibility of $\mathcal{P}_i$.

The CAD scheme is formalized in Figure 2. The core of the algorithm is in lines 3–6 which selects a set of customers (line 3), extracts the customers as a VRPTW (line 4), reoptimizes subproblem $\mathcal{P}_o$ using algorithm $\mathcal{A}$ (line 5), and merges the new optimized subplan $\sigma_o^*$ to obtain the new solution (line 6). These main steps are repeated until the time limit is reached. The extraction step is given by the EXTRACT function, which collects all vehicles serving a customer in the decomposition (line 1), collects all the customers served by these vehicles in between customers of $S$, and constructs the depots (lines 2–10). The customers and depots so obtained define the subproblem (line 11). The CONSTRUCTARRIVALDEPOT and CONSTRUCTDEPARTUREDEPOT functions describe how to create depots for $\mathcal{P}_o$ that allows $\sigma_o^*$ to be feasibly merged into $\sigma$. Finally, the MERGE function shows how $\sigma_o^*$ is merged into $\sigma$.

## 6 Instantiations of the CAD Scheme

This section presents a variety of instantiations of the CAD scheme. Each such instantiation only have to specify how the function SELECTCUSTOMERS is implemented. We start with the vehicle-based spatial decomposition proposed in [4], generalized it, and then presents temporal and random decompositions.

### 6.1 The VASD Scheme

We first show how the VASD scheme can be viewed as an instantiation of CAD. The VASD decomposition scheme is depicted in Figure 3 and aims at choosing wedges

CAD($\mathcal{A}, \sigma_0$)
1 $\sigma \leftarrow \sigma_0$;
2 **while** time limit unreached
3 **do** $S \leftarrow$ SELECTCUSTOMERS($\mathcal{P}, \sigma$);
4 $\mathcal{P}_o \leftarrow$ EXTRACT($S, \mathcal{P}, \sigma$);
5 $\sigma_o^* \leftarrow \mathcal{A}(\mathcal{P}_o)$;
6 $\sigma \leftarrow$ MERGE($\mathcal{P}_o, \sigma_o^*, \sigma$);
7 **return** $\sigma$

EXTRACT($S, \mathcal{P}, \sigma$)
1 $R \leftarrow \{r \in \sigma \mid \exists c \in r : c \text{ lies in } S\}$;
2 $C_o \leftarrow \emptyset$;
3 $D_o^- \leftarrow \emptyset$;
4 $D_o^+ \leftarrow \emptyset$;
5 **for** $r \in R$
6 **do** $i \leftarrow \text{argmin}_{(c \in r \cap S)}\ a_c$;
7 $j \leftarrow \text{argmax}_{(c \in r \cap S)}\ a_c$;
8 $C_o \leftarrow C_o \cup \bigcup_{(c \in r_v): a_i \le a_c \le a_j}$;
9 $D_o^- \leftarrow D_o^- \cup$ CONSTRUCTDEPARTUREDEPOT($pred(i)$);
10 $D_o^+ \leftarrow D_o^+ \cup$ CONSTRUCTARRIVALDEPOT($succ(j)$);
11 **return** $(C_o, \mathcal{D}_o^+, \mathcal{D}_o^-)$;

CONSTRUCTARRIVALDEPOT($p$)
1 $d^- \leftarrow p$;
2 $[e_{d^-}, l_{d^-}] \leftarrow [\delta_p, \infty]$;
3 $q_{d^-} \leftarrow q(p)$;
4 **return** $d^-$;

CONSTRUCTDEPARTUREDEPOT($s$)
1 $d^+ \leftarrow s$;
2 $[e_{d^+}, l_{d^+}] \leftarrow [0, z_s]$;
3 $q_{d^+} \leftarrow q(s) - q(pred(s))$;
4 **return** $d^+$;

MERGE($\mathcal{P}_o, \sigma_o^*, \sigma$)
1 **for** $c \in \mathcal{P}_o$
2 **do** $succ(\sigma, pred(c)) \leftarrow c$;
3 $pred(\sigma, succ(c)) \leftarrow c$;
4 $succ(\sigma, c) \leftarrow succ(\sigma_o^*, c)$;
5 $pred(\sigma, c) \leftarrow pred(\sigma_o^*, c)$;
6 **return** $\sigma$;

**Fig. 2.** The CAD Scheme.

SELECTDECOMPOSITIONVASD($\mathcal{P}, \sigma$)
1 **select** $\alpha \in [0, 359]$;
2 **select** $\beta > \alpha$ **such that** *the wedge* $W \leftarrow (\alpha, \beta)$;
3 (a) contains at least $N$ customers;
4 (b) is the smallest wedge satisfying *(a)*;
5 $\mathcal{V}_t \leftarrow \{v \in \mathcal{V} \mid \exists\, c \in r_v : c \text{ lies in } W\}$;
6 **return** $\bigcup_{v \in \mathcal{V}_t} cust(r_v)$;

**Fig. 3.** The VASD Scheme for VRPTW Decouplings

SELECTDECOMPOSITIONCASD($\mathcal{P}, \sigma$)
1 **select** $\alpha \in [0, 359]$;
2 **select** $\beta > \alpha$ **such that** *the wedge* $W \leftarrow (\alpha, \beta)$;
3 (a) contains at least $N$ customers;
4 (b) is the smallest wedge satisfying *(a)*;
5 **return** $\bigcup c$ lies in $W$;

**Fig. 4.** The CASD Scheme for VRPTW Decouplings

producing roughly the same number $N$ of customers. It first chooses the lower angle $\alpha$ of the wedge randomly (line 1). It then selects the upper angle $\beta$ as the smallest angle greater than $\alpha$ producing the smallest wedge with at least $N$ customers (lines 2–4). Finally, all customers of vehicles within in the wedge are included in the decomposition.

### 6.2 The CASD Scheme

We now present a customer-based spatial decomposition CASD that generalizes the VASD scheme. This generalization is especially important when considering problems (such as the class 2 problems of the extended Solomon benchmarks) where the vehicles serve many customers and can travel across many portions of the space. Under these conditions, VASD loses some of its locality as shown in Figure 1. In contrast, CASD algorithm preserves the spatial boundaries and improves the results of spatial decouplings on the class 2 extended Solomon benchmarks. Figure 4 gives the formalization of CASD which is a simplification of VASD. Figure 5 shows how the CASD scheme performs a decoupling from the same starting solution as Figure 1. The right hand picture shows all routes with decoupled customers, with the decoupled customers shown in red and the remaining ones in blue. It is interesting to compare this with Figure 1. CASD is clearly better at respecting spatial boundaries and allows customers of more vehicles to be considered in the decomposition.

### 6.3 The CATD Scheme

We now present an temporal instantiation (CATD) of the CAT scheme. The CATD scheme chooses random time slices and returns all of the customers that are served within that time slice. Figure 6 provides the implementation of this algorithm where

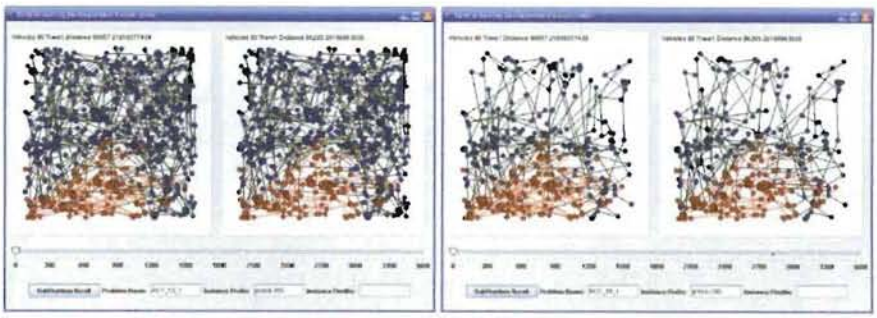


**Fig. 5.** The First Decoupling of CASD.

SELECTDECOMPOSITIONCATD($\mathcal{P}, \sigma$)
1 **select** $\alpha \in [0, l_{d \in \mathcal{D}}]$;
2 **select** $\beta > \alpha$ **such that** the time period $T = (\alpha, \beta)$;
3 (a) contains at least $N$ customers;
4 (b) is the smallest time period satisfying (a);
5 **return** $\bigcup c$ served in $T$;

**Fig. 6.** The CATD Scheme for VRPTW Decouplings

lines 1–4 select a random slice that contains at least $N$ customers. The mechanism for choosing a time period is similar to that of CASD. First, $\alpha$ is chosen randomly from the interval $[0, l_{d \in \mathcal{D}}]$. $\beta$ is then incremented from $\alpha + 1$ until the desired number of customers appear in the interval (or when $\beta = l_{d \in \mathcal{D}}$). Figure 7 demonstrates a decoupling based on the CATD scheme. Unlike prior decouplings, the temporal decoupling crosses most of the vehicles as seen by the number of routes included in the righthand side of the figure.

### 6.4 The CARD Scheme

This section describes a simple random decoupling scheme (CARD) used to provide a basis to evaluate the structured decoupling schemes described in the prior sections. Figure 8 shows the implementation. The scheme iterates by selecting random sequences of customers (lines 3–4) until the desired number of customers is achieved (line 2).

## 7 Experimental Results

This section presents the experimental results for the 1,000 customer extended Solomon benchmarks (`www.top.sintef.no/vrp/benchmarks.html`). The benchmarks contain a mix of loose and tight time windows and different types of spatial distributions. Recall that the difficulty in these problems, once two-stage algorithms are considered, is mostly in optimizing travel distances. Hence the experimental results mostly

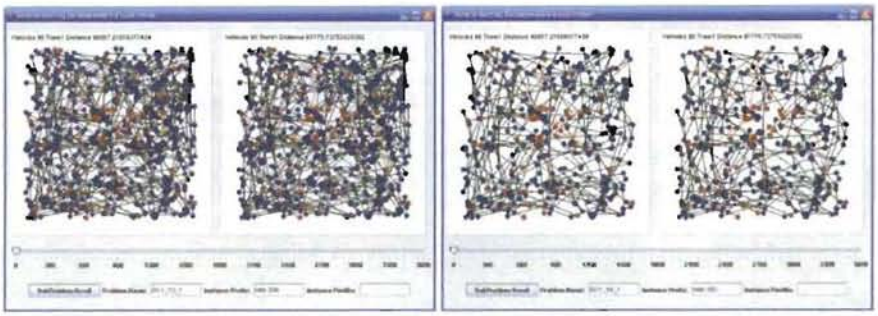


**Fig. 7.** The First Decoupling of CATD.

SELECTDECOMPOSITIONCARD($\mathcal{P}, \sigma$)
1 $S \leftarrow \emptyset$;
2 **while** $|S| < N$
3 **do select** $\alpha \in C \setminus S$;
4 **select** $\beta \in C \setminus S$ such that $route(\alpha) = route(\beta) \wedge \delta_\alpha < \delta_\beta$;
5 $S \leftarrow S \cup \bigcup c \in C$ such that $route(c) = route(\alpha) \wedge \delta_\alpha \leq \delta_c \leq \delta_\beta$;
6 **return** $S$;

**Fig. 8.** The CARD Scheme for VRPTW Decouplings

focus on this second stage, and uses a fixed solution with the minimal number of vehicles from the first phase. The experimental results use large neighborhood search (LNS) [29] for algorithm $\mathcal{A}$. LNS is one of the most effective algorithms for optimizing vehicle routing problems [29, 3, 2, 26, 24]; it also has the benefits of easily accommodating side constraints [3], which is important in practical implementations. The experiments report the solution quality under various time constraints (i.e., 2.5, 5, 10, and 15 minutes). Each reported result is the average of 50 runs on an AMD Athlon Dual Core Processor 3800.

For space reasons, we focus only on class R2. In general, the results on RC1 and R1 show that VASD(LNS) is the best implementation and produces significant improvements in solution quality under time constraints. In average, it produces improvements of 35%, 29%, 17%, and 6% over LNS when the time constraints require solutions to be found within 1, 2.5, 5, and 10 minutes respectively on RC1 problems. Both VASD and CASD outperform LNS on all RC1 and R1 instances and the results of CATD and CARD are good after the first 2.5 minutes. In general, good solutions to RC1 and R1 are characterized by vehicles serving very few customers in narrow regions, making spatial decompositions very natural.

*Benefits of* CAD Table 1 describes the solution quality under various time constraints for LNS and various instantiations of CAD(LNS) on R2 problems. Each column describes a R2 instance with 1,000 customers and the best-known number of vehicles. The

| BK | R2_10_1 | R2_10_2 | R2_10_3 | R2_10_4 | R2_10_5 | R2_10_6 | R2_10_7 | R2_10_8 | R2_10_9 | R2_10_10 | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|
| UB | 42294.31 | 33459.32 | 24938.95 | 17880.11 | 36258.34 | 30073.6 | 23253.89 | 17509.69 | 33068.74 | 30312.5 | |
| LNS (1) | 56336.2 | 43864.4 | 42620.2 | 33281.9 | 47352.4 | 40907.5 | 38056.6 | 29516.6 | 44540.9 | 40973.8 | |
| VASD (1) | 67937.5 | 48391.3 | 47151.0 | 29349.6 | 59075.1 | 49561.9 | 38104.1 | 26981.8 | 55860.8 | 47982.2 | |
| %Impr. | -20.6 | -10.3 | -10.6 | 11.8 | -24.8 | -21.2 | -0.1 | 8.6 | -25.4 | -17.1 | -11.0 |
| CASD (1) | 68108.7 | 50337.8 | 47611.1 | 28388.1 | 56962.9 | 50430.3 | 37210.3 | 25543.2 | 56308.1 | 51254.3 | |
| %Impr. | -20.9 | -14.8 | -11.7 | 14.7 | -20.3 | -23.3 | 2.2 | 13.5 | -26.4 | -25.1 | -11.2 |
| CATD (1) | 51346.6 | 42352.3 | 41823.3 | 34448.2 | 46275.7 | 43460.7 | 38282.0 | 31186.8 | 42321.8 | 42122.2 | |
| %Impr. | 8.9 | 3.4 | 1.9 | -3.5 | 2.3 | -6.2 | -0.6 | -5.7 | 5.0 | -2.8 | 0.3 |
| CARD (1) | 76915.0 | 63039.1 | 57772.6 | 40365.4 | 59084.6 | 60321.9 | 48678.5 | 35275.7 | 66420.1 | 62966.8 | |
| %Impr. | -36.5 | -43.7 | -35.6 | -21.3 | -24.8 | -47.5 | -27.9 | -19.5 | -49.1 | -53.7 | -36.0 |
| LNS (2.5) | 53667.5 | 41260.3 | 37907.6 | 30007.5 | 44941.5 | 38028.4 | 33939.7 | 26921.2 | 42134.0 | 38351.7 | |
| VASD (2.5) | 58759.6 | 41955.8 | 38316.4 | 24632.8 | 49847.5 | 38975.1 | 32055.8 | 23029.9 | 46152.3 | 40896.1 | |
| %Impr. | -9.5 | -1.7 | -1.1 | 17.9 | -10.9 | -2.5 | 5.6 | 14.5 | -9.5 | -6.6 | -0.4 |
| CASD (2.5) | 54423.3 | 40426.1 | 33387.2 | 22717.8 | 46063.0 | 38462.0 | 29460.6 | 20837.6 | 43235.2 | 39663.2 | |
| %Impr. | -1.4 | 2.0 | 11.9 | 24.3 | -2.5 | -1.1 | 13.2 | 22.6 | -2.6 | -3.4 | 6.3 |
| CATD (2.5) | 46203.4 | 38061.9 | 33749.0 | 28799.4 | 40220.7 | 36499.3 | 32848.3 | 26997.1 | 37653.3 | 34791.7 | |
| %Impr. | 13.9 | 7.8 | 11.0 | 4.0 | 10.5 | 4.0 | 3.2 | -0.3 | 10.6 | 9.3 | 7.4 |
| CARD (2.5) | 65820.0 | 50880.9 | 43792.2 | 31651.3 | 56636.9 | 47119.1 | 37598.2 | 25336.4 | 54375.1 | 50583.8 | |
| %Impr. | -22.6 | -23.3 | -15.5 | -5.5 | -26.0 | -23.9 | -10.8 | 5.9 | -29.1 | -31.9 | -18.3 |
| LNS (5) | 51877.8 | 39871.7 | 34873.2 | 27549.9 | 43616.4 | 36400.2 | 31500.3 | 25323.0 | 40647.4 | 37109.6 | |
| VASD (5) | 54743.7 | 40546.3 | 34540.3 | 22899.2 | 46174.3 | 36959.9 | 30188.5 | 21775.7 | 42417.0 | 38351.2 | |
| %Impr. | -5.5 | -1.7 | 1.0 | 16.9 | -5.9 | -1.5 | 4.2 | 14.0 | -4.4 | -3.3 | 1.4 |
| CASD (5) | 49454.3 | 38194.8 | 30138.7 | 21578.2 | 42203.5 | 34796.8 | 27451.5 | 20837.6 | 38577.2 | 35847.8 | |
| %Impr. | 4.9 | 4.4 | 15.7 | 27.7 | 3.3 | 4.6 | 14.7 | 21.5 | 5.4 | 3.5 | 10.6 |
| CATD (5) | 44633.9 | 36339.7 | 31647.7 | 26463.1 | 39040.4 | 34816.3 | 29354.2 | 25357.9 | 36014.8 | 33517.4 | |
| %Impr. | 14.0 | 8.9 | 9.2 | 3.9 | 10.5 | 4.4 | 6.8 | -0.1 | 11.4 | 9.7 | 7.9 |
| CARD(5) | 58595.0 | 44269.4 | 37808.6 | 27458.0 | 49889.9 | 40801.7 | 32854.8 | 23217.5 | 47379.4 | 43606.2 | |
| %Impr. | -5.5 | -1.7 | 1.0 | 16.9 | -5.9 | -1.5 | 4.2 | 14.0 | -4.4 | -3.3 | 1.4 |
| LNS (10) | 50763.2 | 38737.0 | 34873.2 | 25195.6 | 42848.5 | 35342.0 | 29752.8 | 23665.7 | 39802.5 | 36378.8 | |
| VASD (10) | 51950.6 | 39427.7 | 32426.1 | 22185.2 | 44327.2 | 35842.8 | 29264.1 | 21164.4 | 40519.9 | 37099.5 | |
| %Impr. | -2.3 | -1.8 | 7.0 | 11.9 | -3.5 | -1.4 | 1.6 | 10.6 | -1.8 | -2.0 | 1.8 |
| CASD (10) | 47371.3 | 37343.2 | 28991.6 | 21010.5 | 40890.2 | 33852.4 | 26566.4 | 20290.5 | 37112.5 | 34632.8 | |
| %Impr. | 6.7 | 3.6 | 16.9 | 16.6 | 4.6 | 4.2 | 10.7 | 14.3 | 6.8 | 4.8 | 8.9 |
| CATD (10) | 44172.2 | 36339.7 | 31358.7 | 25469.9 | 38445.3 | 33830.0 | 29354.2 | 24371.9 | 35221.8 | 32786.1 | |
| %Impr. | 13.0 | 6.2 | 10.1 | -1.1 | 10.3 | 4.3 | 1.3 | -3.0 | 11.5 | 9.9 | 6.2 |
| CARD (10) | 52845.8 | 40408.6 | 33549.5 | 24331.5 | 45462.6 | 36987.2 | 29852.5 | 23217.5 | 42537.3 | 38638.1 | |
| %Impr. | -4.1 | -4.3 | 3.8 | 3.4 | -6.1 | -4.7 | -0.3 | 1.9 | -6.9 | -6.2 | -2.3 |

**Table 1.** R2 Solution Quality Under Time Constraints.

clusters of rows consider various time constraints: 1, 2.5, 5, and 10 minutes. The row $BK$ specifies the travel distance of the best known solution (prior to this research). The rows %Impr describes the improvement in solution quality of CAD(LNS) with respect to LNS. CAD(LNS) is run with $N = 200$, i.e., the decomposition must contain at least 200 customers.

It is interesting to observe that Table 1 provides very different conclusions than the results on classes RC1 and R1. High-quality solutions to R2 problems are characterized by fewer vehicles serving many more customers over wide temporal regions. This puts VASD at a disadvantage as decompositions typically violate the natural spatial boundaries of the wedge due to the need to include all customers of vehicles. This is best illustrated by the 5 minute results, when the CASD scheme vastly outperforms VASD. After 2.5, 5, and 10 minutes, CASD produces average improvements of 6.3%, 10.6%, and 8.9% over LNS, while VASD degrades the performance after 2.5 minutes and produces improvements of 1.4% and 1.6% for 5 and 10 minutes. On some benchmarks, CASD produces more than 10% over LNS. Interestingly, CATD produces excellent re-

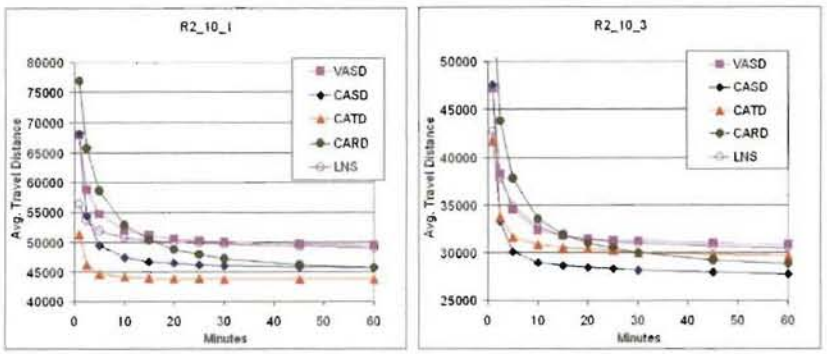



**Fig. 9.** Benefits of CAD on R2_10_1 and R2_10_3.

sults on class R2 and produces average improvements of 0.3%, 7.5%, 7.9%, and 6.2% after 1, 2.5, 5, and 10 minutes. Moreover, it significantly outperforms other decompositions on several benchmarks where it can produce improvements up to 14%. On closer inspection, CATD performs very well on those problems whose customers have narrow time windows. The explanation for this behavior is interesting: when a customer has a wide time window, it can be served early or late. If it is initially served early when it should be served late, it is impossible to find a solution that moves the customer to a later time period, unless every intermediate temporal decoupling provides an improving solution. On problems with customers with narrow time windows, the problem structure itself enforces the correct temporal locations of the customers, making a temporal decomposition very natural.

Figure 9 depicts the typical behavior of LNS and CAD(LNS) on two benchmarks in the R2 class. In the left graph, the R2 problem has narrow time windows and CATD is clearly the best, further demonstrating the natural benefits of this decomposition when customers have narrow time windows. It also shows the limitations of the VASD approach under the conditions of class 2 problems. The right part of the figure shows results on a class 2 problem with wide time windows. Here we see a reversal of the effectiveness of CATD where CASD is clearly better. Note also that CASD(LNS) and CATD(LNS) still dominates LNS when both algorithms run for an hour.

Overall, these results clearly show the benefits of customer-based decompositions and the complementary between spatial and temporal decompositions.

*Hybrid Implementations* To exploit this complementarity, We also considered some hybrid approaches between CASD and CATD to determine if a single approach would perform well on all instances (for example good on both R2_10_1 and R2_10_3). Two hybrids worked quite well. The first hybrid chooses to either follow a CATD decoupling or a CASD decoupling randomly at each iteration. The second hybrid creates a decoupling at each iteration that contains $N/2$ customers from a CATD selection and $N/2$ from CASD selection. Both schemes generated very consistent results on all problems, in general being within 1% of the best CASD or CATD result on each problem.

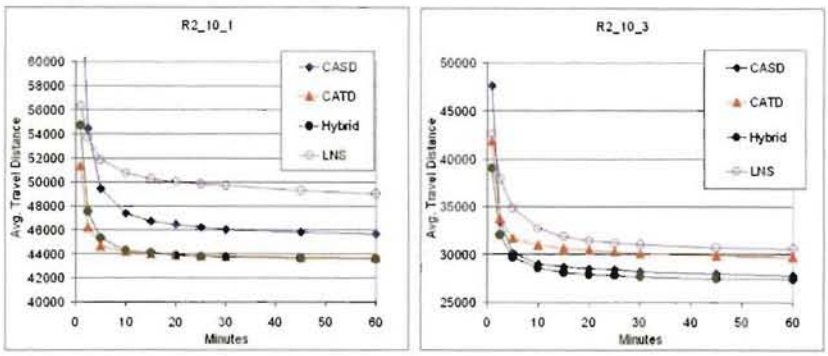



**Fig. 10.** Benefits of Hybrid Approaches.

This indicates that when problem structure is unknown or varied, a hybrid approach may produce the best results. Figure 10 demonstrates how hybrid based approaches can smooth out performance.

## 8 Related Work

There are literally hundreds of papers discussing vehicle routing problems and their variations and it is beyond the scope of this paper to provide a comprehensive literature review. The reader is invited to see [8, 9, 11, 16, 27, 25] for recent surveys. Almost all papers focus on problems of relatively small size which, as mentioned earlier, are already extremely difficult. Unfortunately, many of the proposed techniques do not scale well and some recent papers specifically address large-scale problems. We now focus attention on recent work that have considered decomposition ideas.

Decomposition comes in many different varieties in literature. In some papers, like [5, 6], decomposition focuses on decomposing the search strategy space (as opposed to problem structure). Related to this idea is the view of decomposition across attributes (variables) of the problems. Multi-stage approaches such as [15, 2, 21, 18, 10, 7] can be classified in this way (i.e., first minimizing the number vehicles required and then minimizing the travel distance). [12] suggests a general framework for breaking problems across attribute boundaries using evolutionary algorithms. The different subproblems communicate results via population exchanges. The framework is tested on the VRPTW. They key difference between attribute decomposition and CAD is that our approach retains information about the entire problem and simplifies the problem by decreasing their scale.

Recent and concurrent work has focused on dividing the problem into smaller subproblems across structural boundaries that is very much in the spirit of VASD. [20] presents a deterministic hierarchical decomposition scheme for evolutionary algorithms. The VRPTW spatial region is divided into rectangles, defining sub problems that are solved independently. The rectangles are recursively merged into larger subproblems

which rely on the smaller problems as starting solutions for the larger subproblems. [1] introduces spatial-based decomposition ideas in a genetic algorithm. Their approach randomly applies the evolutionary operations to either the whole problem or spatially defined sub regions. [23, 22] presents some interesting spatial decomposition approaches based on clustering (POPMUSIC). At a high level, POPMUSIC iteratively chooses routes and creates subproblems based on *nearness* to that route (different approaches to defining nearness are explored). The algorithm iterates until it has created subproblems on all routes without improvement. Finally, the work of [13] proposes a decoupling scheme for the air-taxi problem based on spatial boundaries.

In many ways, all of these approach can be viewed as variations of VASD. *The key difference between these approaches and our framework is that they decompose problems based on routes as opposed to customers.* This makes the merging of solutions from the subproblems to the global problem easy. However, by structuring the decompositions on a customer basis, we are able to create subproblems *within* routes, a property that is very important when routes cross multiple spatial and temporal boundaries. But is important to note that this related work also supports our claim that decomposition improves algorithm performance.

It is useful to contrast the deconstruction steps of LNS ([29, 26, 28, 24]) and the CAD scheme. In LNS, the basic step consists of removing related customers (often based on spatial or temporal relationships) from a plan $\sigma$ and to reinsert them in $\sigma$ using an optimization algorithm. The CAD scheme can also be thought of as removing related customers with two fundamental differences: *1) the removed customers defines a VRPTW subproblem of (significantly) smaller size which can solved independently and 2) Subproblems restrict neighborhood explorations to being within the decomposition itself.* This is critical for finding high-quality solution quickly. Obviously, the two approaches are synergetic since our results are obtained using CAD(LNS).

Finally, it is useful to relate CAD to the approach in [17] which impose specific temporal constraints to obtain decouplings. CAD uses spatial and temporal decouplings that constrain specific subsets of customers to be served by designated vehicles. Moreover, the use of decoupling is fundamentally different. *The idea is to iteratively obtain new decouplings to optimize an existing plan by re-optimizing subproblems.* This use of decouplings also contrast with traditional decomposition techniques in constraint satisfaction [14].

## 9 Conclusion

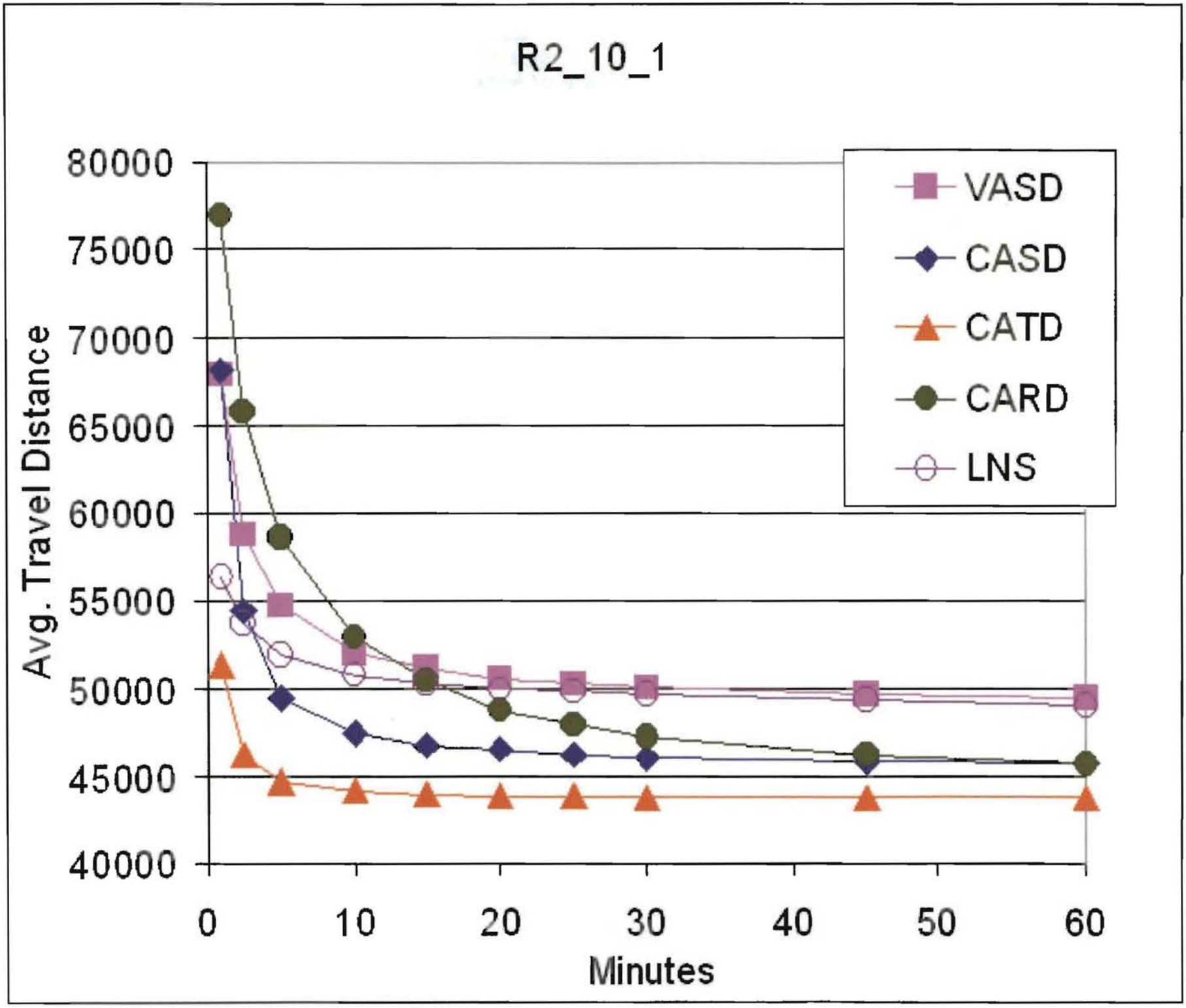
This paper reconsidered the adaptive decomposition framework to quickly find high-quality solutions to large-scale vehicle routing problems with time windows. Earlier work had focused vehicle-based decompositions that partition the vehicles across the subproblems which makes it easy to define the subproblems and merge their solutions. Although vehicle-based spatial decompositions are very effective on classes R1 and RC1 of the extended Solomon benchmarks, the paper identified some of their limitations and, in particular, the difficulty in adapting them to temporal decompositions. This paper then proposed customer-based decompositions which generalize vehicle-based decouplings and allow for focused spatial and temporal decompositions. Experimen-

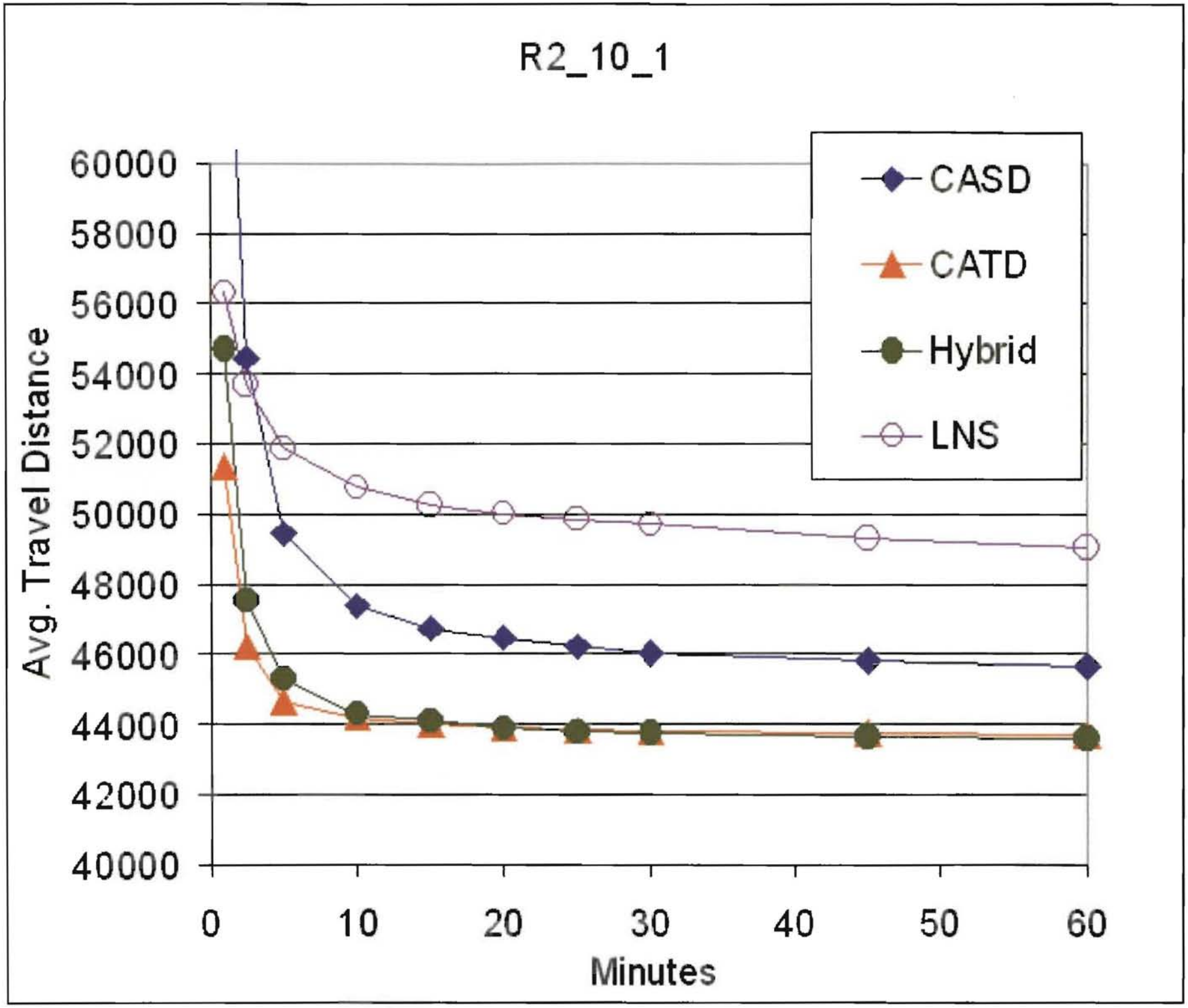
tal results on class R2 of the extended Solomon benchmarks demonstrated the benefits of the customer-based adaptive decomposition scheme and its spatial, temporal, and hybrid instantiations. In particular, the results show significant benefits over the use of large neighborhood search and vehicle-based spatial decompositions. For instance, customer-based temporal decompositions yield an average improvement of 7.4% over LNS after 2.5 minutes, while the vehicle-based spatial decomposition degrades the performance by 0.4% in average. Similarly, customer-based spatial decompositions yield an average improvement of 10.6% over LNS after 5 minutes, while the vehicle-based spatial decomposition improves the performance by only 1.4% in average. The complementary between spatial and temporal decompositions was also highlighted and hybridizations were shown to be particularly effective in producing robust results across all benchmarks.

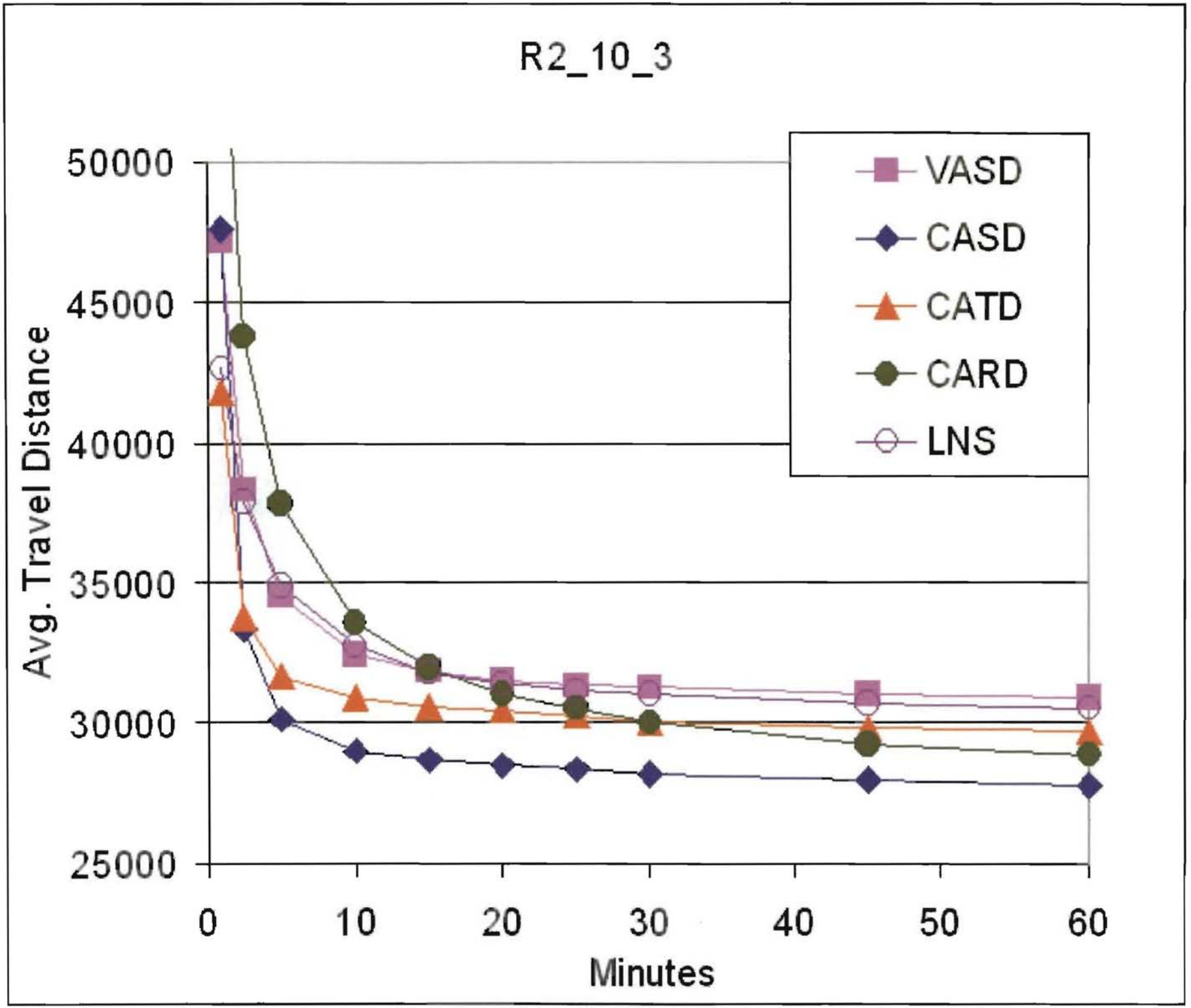
## References

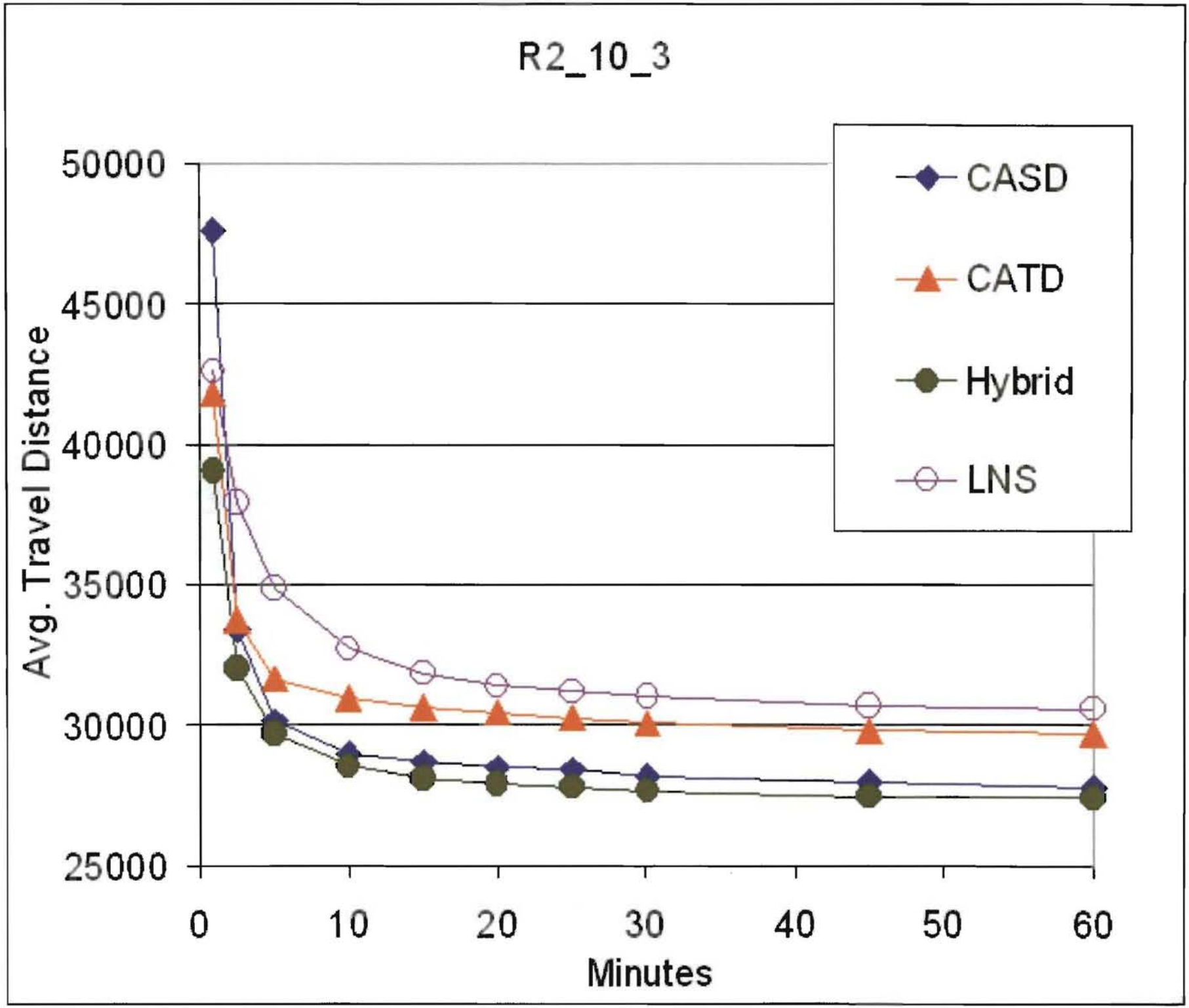

1. G.B. Alvarenga, G.R. Mateus, and G. de Tomi. A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research*, 34:1561–1584, 2007.
2. Russell Bent and Pascal Van Hentenryck. A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows. *Transportation Science*, 38(4):515–530, 2004.
3. Russell Bent and Pascal Van Hentenryck. A Two-Stage Hybrid Algorithm for Pickup and Delivery Vehicle Routing Problems with Time Windows. *Computers and Operations Research*, 33 (4):875–893, 2006.
4. Russell Bent and Pascal Van Hentenryck. Randomized Adaptive Spatial Decoupling For Large-Scale Vehicle Routing with Time Windows. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI)*, Vancouver, Canada, 2007.
5. Alexandre Le Bouthillier and Teodor Crainic. A Cooperative Parallel Meta-Heuristic for the Vehicle Routing Problem with Time Windows. *Computers and Operations Research*, 32:1685–1708, 2005.
6. Alexandre Le Bouthillier, Teodor Crainic, and Peter Kropf. A Guided Cooperative Search for the Vehicle Routing Problem with Time Windows. *IEEE Intelligent Systems*, 20 (4):36–42, 2005.
7. Olli Braysy. A Reactive Variable Neighborhood Search for the Vehicle Routing Problem with Time Windows. *INFORMS Journal on Computing*, 15(4):347–368, 2003.
8. Olli Braysy and Michel Gendreau. Vehicle Routing Problems with Time Windows, Part i: Route Construction and Local Search Algorithms. *Transportation Science*, 39:104–118, 2005.
9. Olli Braysy and Michel Gendreau. Vehicle Routing Problems with Time Windows, Part ii: Metaheuristics. *Transportation Science*, 39:119–139, 2005.
10. Olli Braysy, Geir Hasle, and Wout Dullaert. A Multi-Start Local Search for the Vehicle Routing Problem with Time Windows. *European Journal of Operational Research*, 159 (3):586–605, 2004.
11. Jean-Francois Cordeau, G. Desaulniers, Jacques Desrosiers, Marius Solomon, and Francois Soumis. The VRP with Time Windows. *The Vehicle Routing Problem: SIAM Monographs on Discrete Mathematics and Applications*, pages 157–194, 2001.
12. Teodor Gabriel Crainic, Gloria Cerasela Crisan, Michel Gendreau, Nadia Lahrichi, and Walter Rei. A concurrent evolutionary approach for rich combinatorial optimization. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference (GECCO 09)*, pages 2017–2022, New York, NY, USA, 2009. ACM.

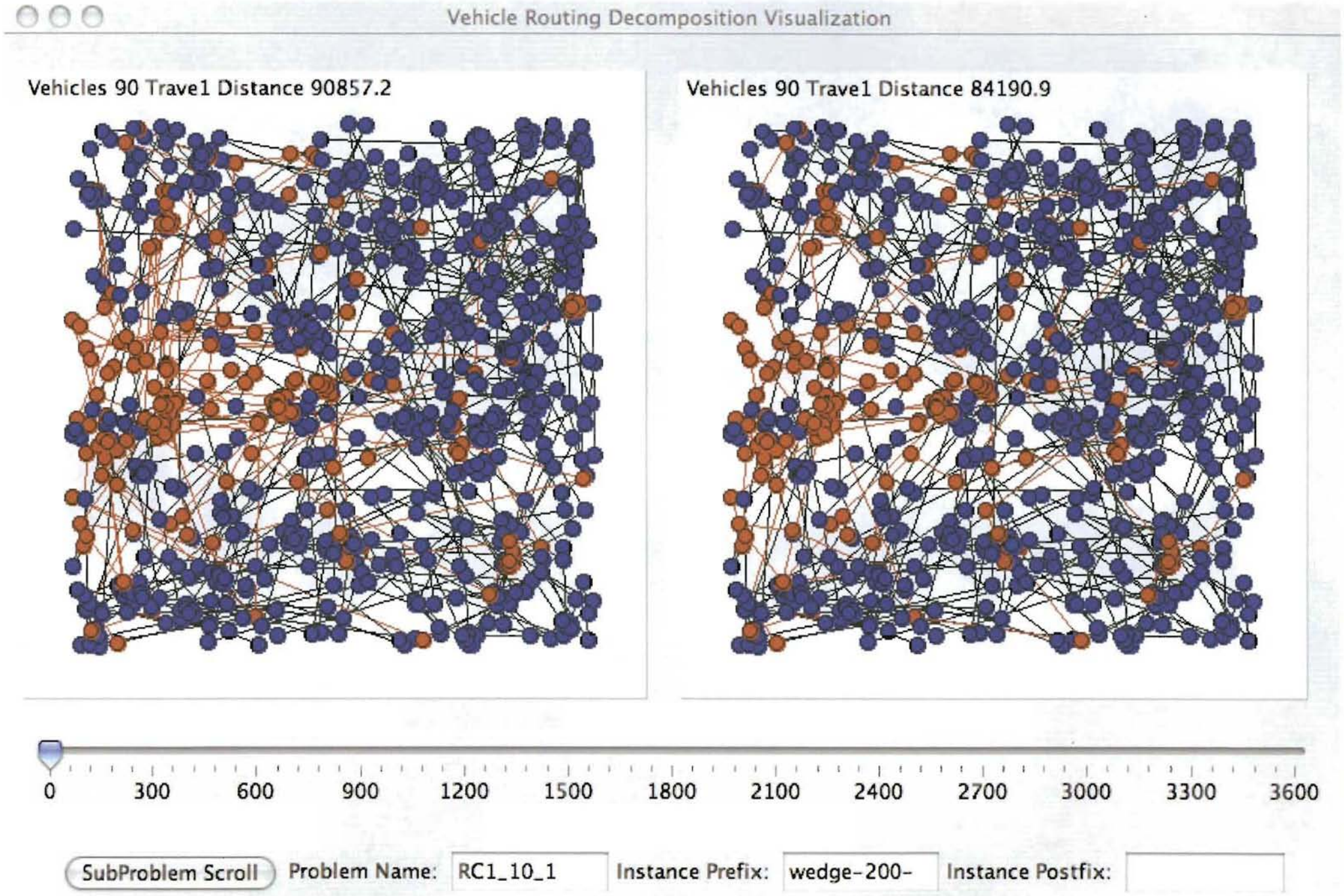
13. D. Espinoza D, R. Garcia, M. Goycoolea, G.L. Nemhauser, , and M. Savelsbergh. Per-Seat, On-Demand Air Transportation Part II: Parallel Local Search. *Transportation Science*, 42 (3):279–291, 2008.
14. Rita Dechter. *Constraint Processing*. Morgan Kaufmann, 2003.
15. Hermann Gehring and Jorg Homberger. A Parallel Two-phase Metaheuristic for Routing Problems with Time Windows. *Asia-Pacific Journal of Operational Research*, 18:35–47, 2001.
16. Bruce Golden, S. Raghavan, and Edward Wasil. *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, 2008.
17. Luke Hunsberger. Algorithms for a Temporal Decoupling Problem in Multi-Agent Planning. In *Proceedings of the Eighteenth American Conference on Artifical Intelligence (AAAI)*, pages 468–475, Edmonton, Canada, 2002.
18. Andrew Lim and Xingwen Zhang. A two-stage heuristic with ejection pools and generalized ejection chains for the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 19(3):443–457, 2007.
19. David Mester and Olli Braysy. Active Guided Evolution Strategies for Large Scale Vehicle Routing Problems with Time Windows. *Computers and Operations Research*, 32:1593–1614, 2005.
20. David Mester, Olli Braysy, and Wout Dullaert. A multi-parametric evolution strategies algorithm for vehicle routing problems. *Expert Systems with Applications*, 32:508–517, 2007.
21. Yuichi Nagataa, Olli Brysy, and Wout Dullaert. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers and Operations Research*, 37 (4):724–737, 2010.
22. Alexander Ostertag, Karl Doerner, Richard Hartl, Eric Taillard, and Philippe Waelti. POPMUSIC for a real-world large-scale vehicle routing problem with time windows. *Journal of the Operational Research Society*, 60(7):934–943, 2009.
23. Alexander Ostertag, Karl F. Doerner, and Richard F. Hartl. A variable neighborhood search integrated in the popmusic framework for solving large scale vehicle routing problems. In *HM '08: Proceedings of the 5th International Workshop on Hybrid Metaheuristics*, pages 29–42, Berlin, Heidelberg, 2008. Springer-Verlag.
24. David Pisinger and Stefan Ropke. Large neighborhood search. In Jean-Yves Potvin and Michel Gendreau, editors, *Handbook of Metaheuristics*. Springer-Verlag, 2009.
25. Jean-Yves Potvin. A review of bio-inspired algorithms for vehicle routing. In Francisco Baptista Pereira and Jorge Tavares, editors, *Studies in Computational Intelligence*, chapter 1, pages 1–34. Springer, 2008.
26. Eric Prescott-Gagnon, Guy Desaulniers, and Louis-Martin Rousseau. A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. *Networks*, pages 1–15, to appear.
27. Panagiotis P. Repoussis, Christos D. Tarantilis, and George Ioannou. Arc-guided evolutionary algorithm for the vehicle routing problem with time windows. *IEEE Transactions on Evolutionary Computation*, 13(3):624–647, 2009.
28. Louis-Martin Rousseau, Michel Gendreau, and Gilles Pesant. Using Constraint-Based Operators to Solve the Vehicle Routing Problem with Time Windows. *Journal of Heuristics*, 8(1):43–58, 2002.
29. Paul Shaw. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In *Proceedings of the Fourth International Conference on the Principles and Practice of Constraint Programming (CP)*, pages 417–431, Pisa, Italy, 1998.

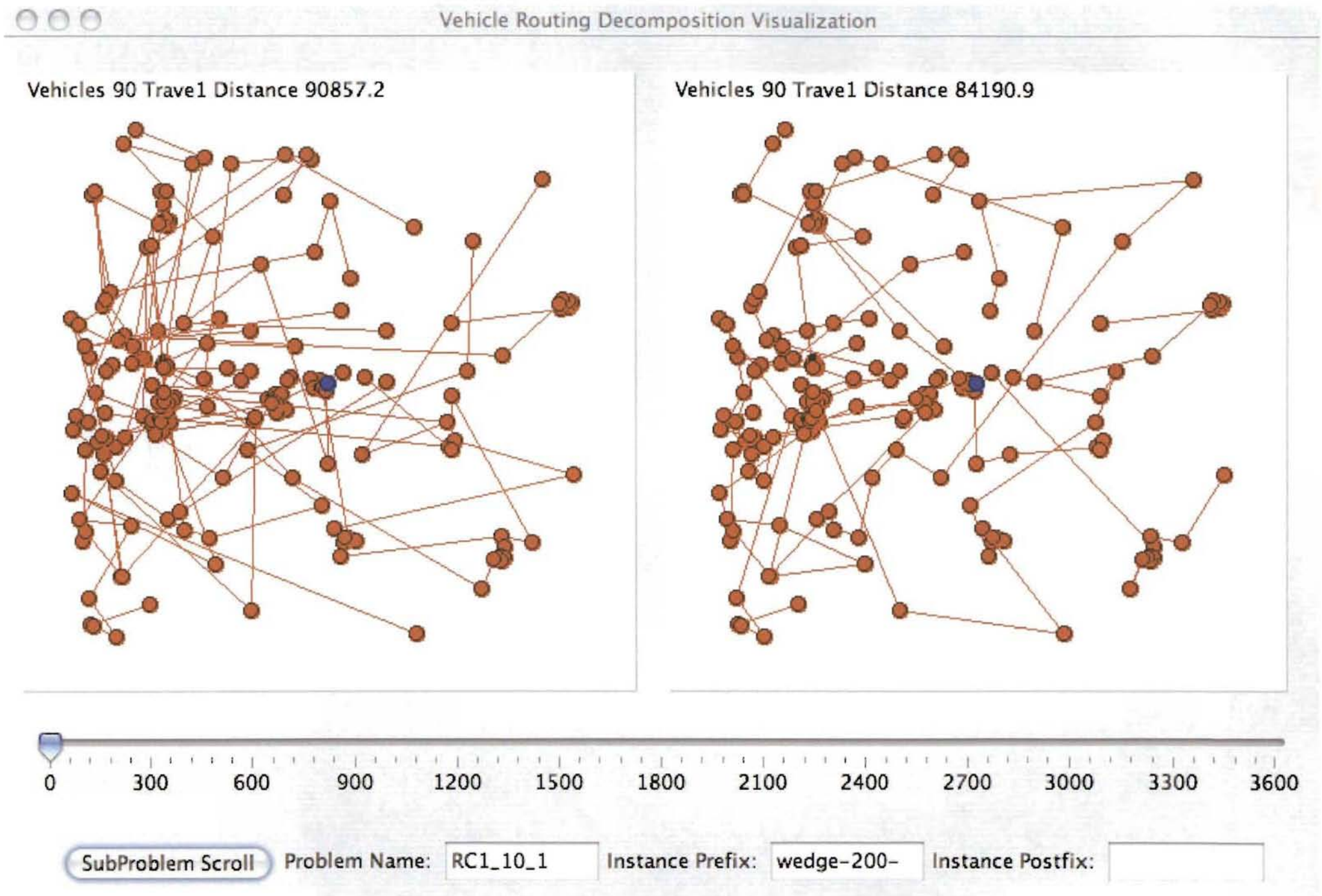

R2_10_1
80000
75000
70000
65000
60000
55000
50000
45000
40000
Avg. Travel Distance
0
10
20
30
40
50
60
Minutes
VASD
CASD
CATD
CARD
LNS

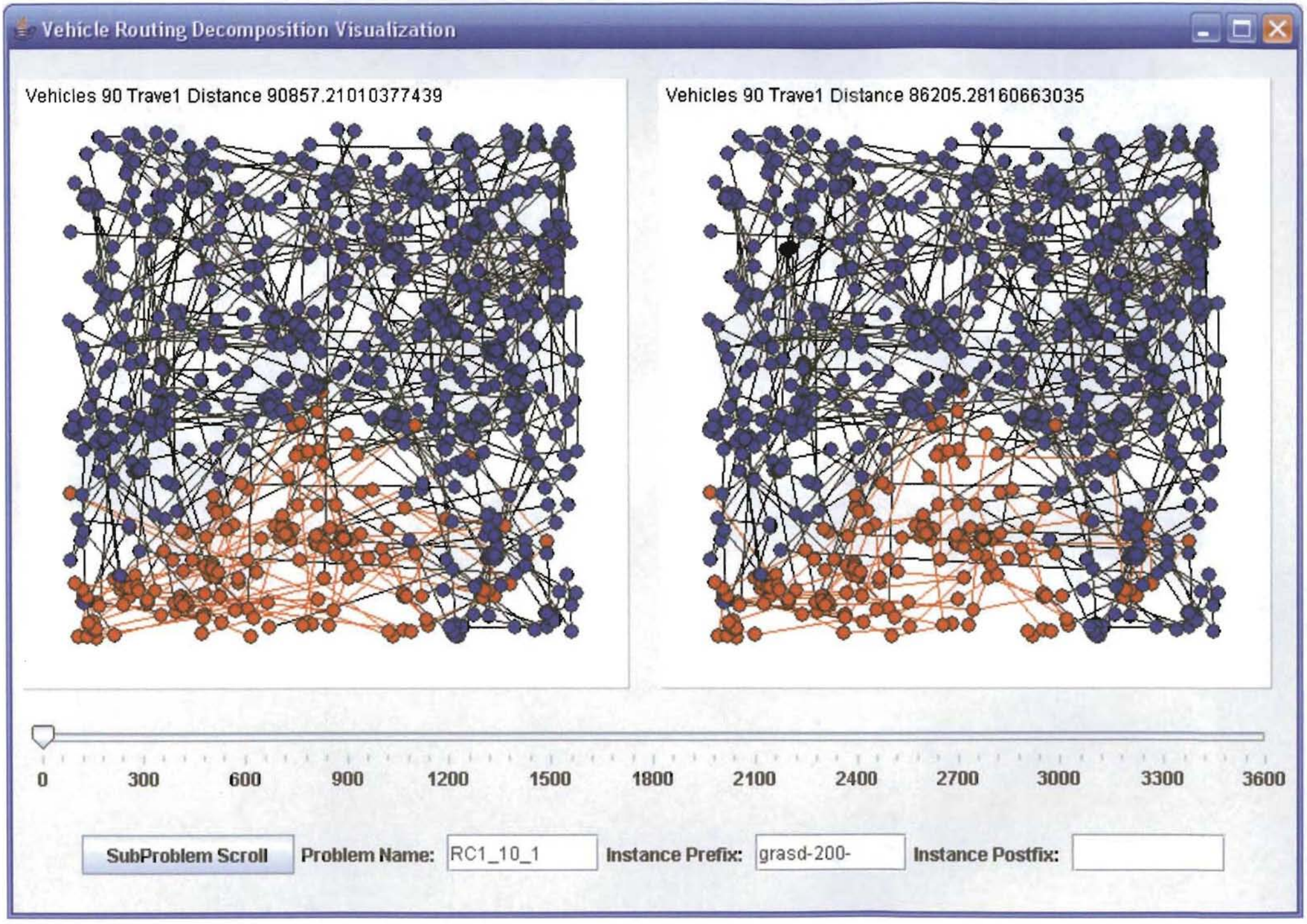

R2_10_1
Avg. Travel Distance
60000
58000
56000
54000
52000
50000
48000
46000
44000
42000
40000
0
10
20
30
40
50
60
Minutes
CASD
CATD
Hybrid
LNS

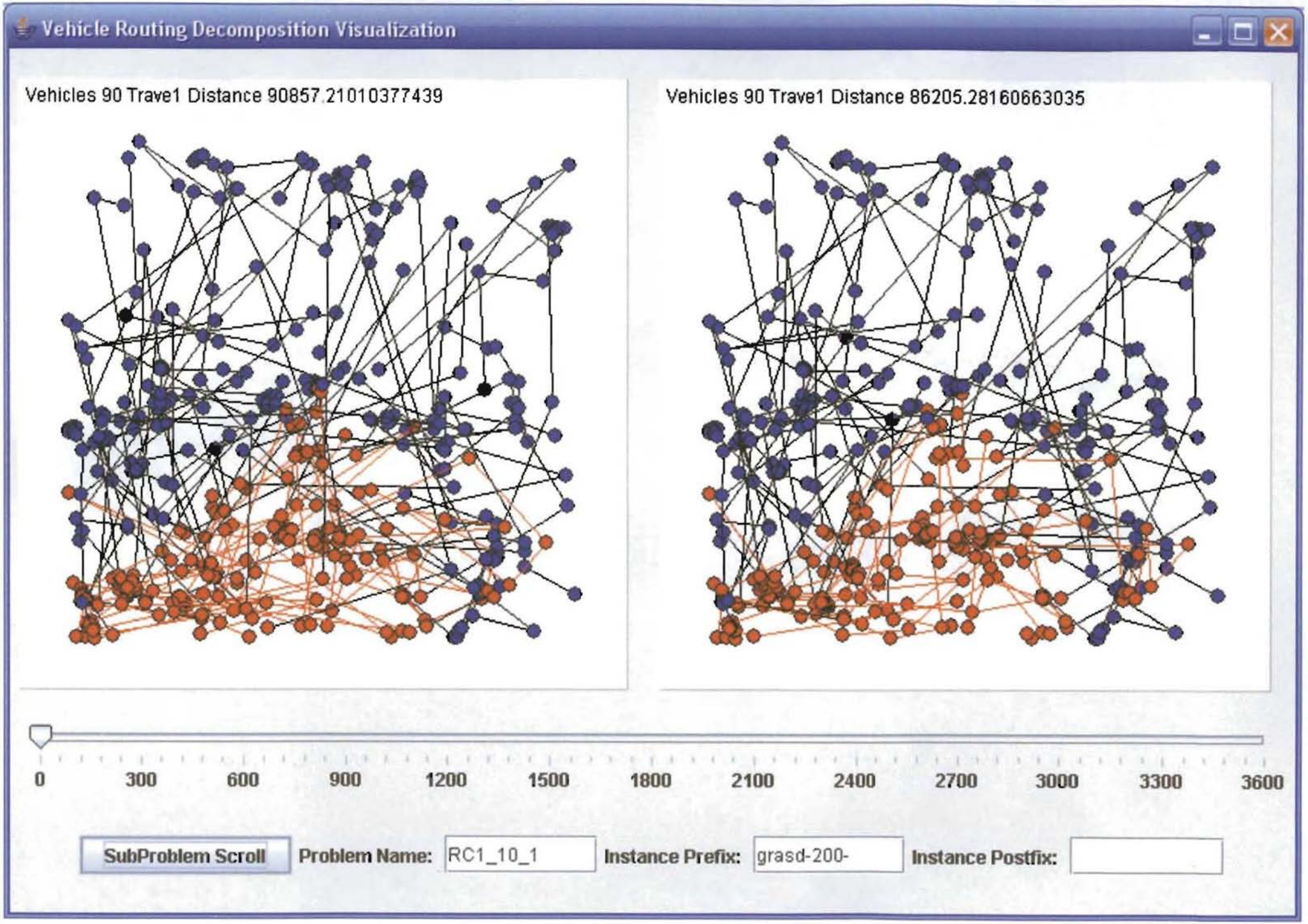

R2_10_3
50000
45000
40000
35000
30000
25000
Avg. Travel Distance
0
10
20
30
40
50
60
Minutes
VASD
CASD
CATD
CARD
LNS

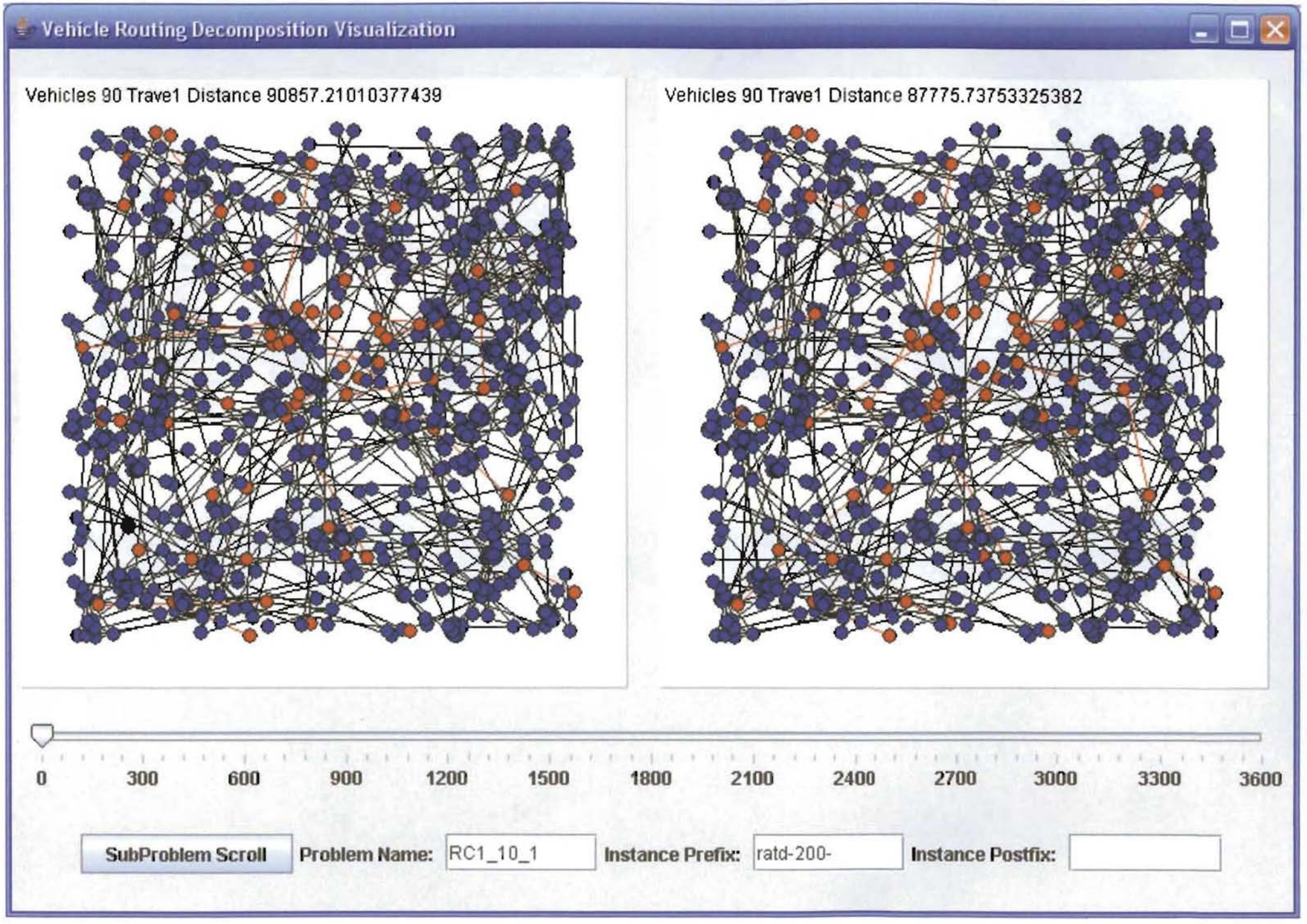

R2_10_3
50000
45000
40000
35000
30000
25000
Avg. Travel Distance
0
10
20
30
40
50
60
Minutes
CASD
CATD
Hybrid
LNS

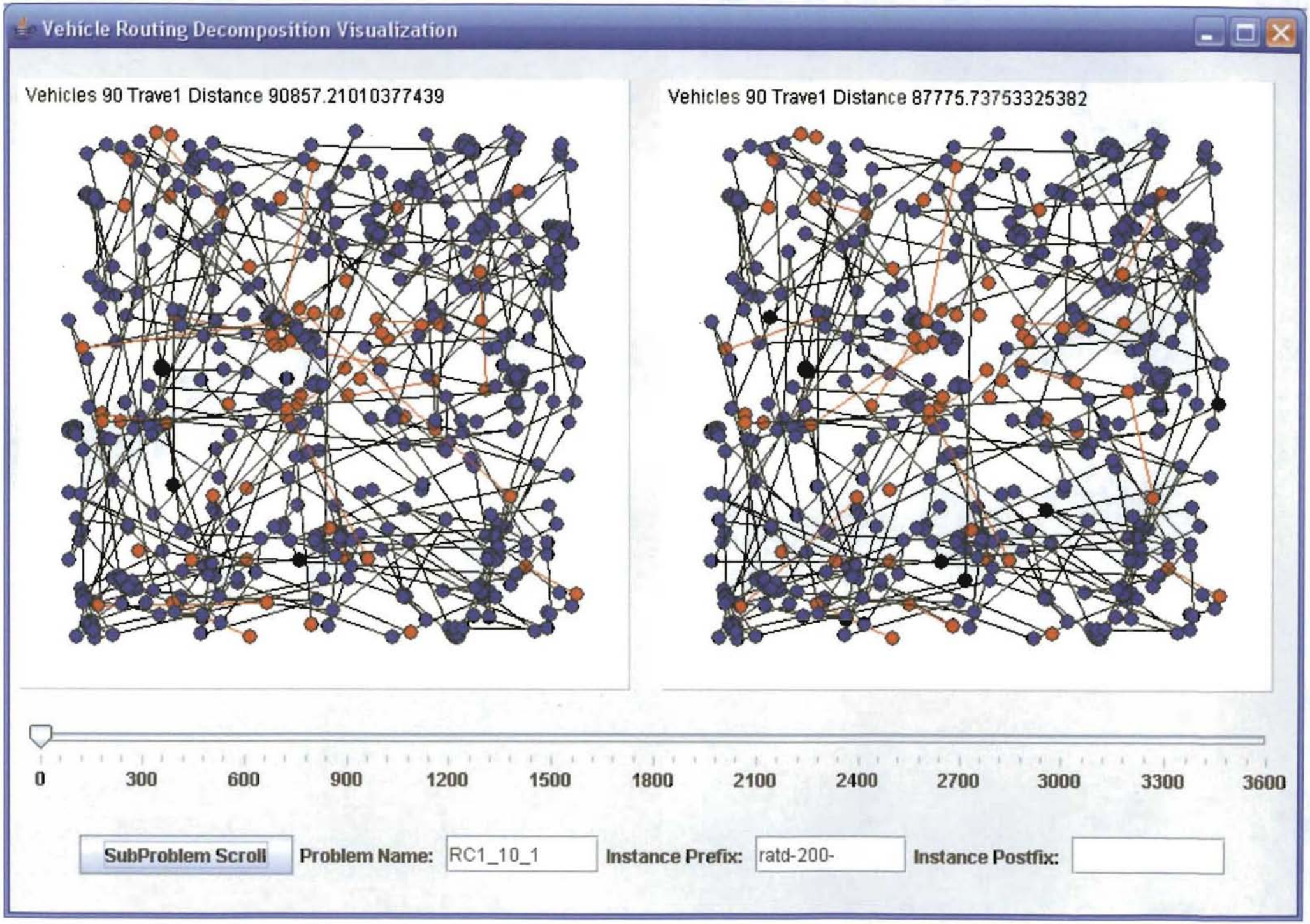

Vehicle Routing Decomposition Visualization
Vehicles 90 Travel Distance 90857.2
Vehicles 90 Travel Distance 84190.9
0
300
600
900
1200
1500
1800
2100
2400
2700
3000
3300
3600
SubProblem Scroll
Problem Name:
RC1_10_1
Instance Prefix:
wedge-200-
Instance Postfix:


Vehicle Routing Decomposition Visualization
Vehicles 90 Travel Distance 90857.2
Vehicles 90 Travel Distance 84190.9
0
300
600
900
1200
1500
1800
2100
2400
2700
3000
3300
3600
SubProblem Scroll
Problem Name:
RC1_10_1
Instance Prefix:
wedge-200-
Instance Postfix:


Vehicle Routing Decomposition Visualization
Vehicles 90 Trave1 Distance 90857.21010377439
Vehicles 90 Trave1 Distance 86205.28160663035
0
300
600
900
1200
1500
1800
2100
2400
2700
3000
3300
3600
SubProblem Scroll
Problem Name:
RC1_10_1
Instance Prefix:
grasd-200-
Instance Postfix:


Vehicle Routing Decomposition Visualization
Vehicles 90 Trave1 Distance 90857.21010377439
Vehicles 90 Trave1 Distance 86205.28160663035
0
300
600
900
1200
1500
1800
2100
2400
2700
3000
3300
3600
SubProblem Scroll
Problem Name:
RC1_10_1
Instance Prefix:
grasd-200-
Instance Postfix:


Vehicle Routing Decomposition Visualization
Vehicles 90 Trave1 Distance 90857.21010377439
Vehicles 90 Trave1 Distance 87775.73753325382
0
300
600
900
1200
1500
1800
2100
2400
2700
3000
3300
3600
SubProblem Scroll
Problem Name:
RC1_10_1
Instance Prefix:
ratd-200-
Instance Postfix:


Vehicle Routing Decomposition Visualization
Vehicles 90 Trave1 Distance 90857.21010377439
Vehicles 90 Trave1 Distance 87775.73753325382
0
300
600
900
1200
1500
1800
2100
2400
2700
3000
3300
3600
SubProblem Scroll
Problem Name:
RC1_10_1
Instance Prefix:
ratd-200-
Instance Postfix: