Multi-agent Systems

Toon Nolten, Joren Verspeurt

March 2014

1 Introduction

We have tried to find a solution to the Pickup and Delivery Problem with dynamic Time Windows (PDPTW), using a Multi-Agent Systems approach. We started off by designing a naive algorithm: an algorithm that does not take into account the presence of any other agents and does not plan for the future but only for the current action. Then we implemented a more complex algorithm, based on some principles we deemed useful, e.g. agents should spread out not cluster together, agents should communicate and for the system to be practical agents should have a limited range of sight and communication. To compare these algorithms we executed them on the Gendreau et al. (2006) dataset.

1.1 Objectives

1. The final MAS has lower lateness and overtime in every tested instance.
2. The final MAS has no properties that might hinder adoption as a real world algorithm, i.e. it cannot rely on global knowledge, communication between agents is not 100% reliable, etc.
3. The final MAS needs to be easily adaptable to different environments.

1.2 Questions

1. (a) How does the frequency of parcel announcements influence lateness and overtime?
   (b) How does the number of agents influence lateness and overtime?
   (c) Can more complex behaviour, i.e. taking into account more variables, decrease lateness and overtime?
2. (a) How does performance depend on the amount of knowledge an agent is able to acquire about his surroundings?
   (b) What is a reasonable communication range between agents? Reasonable means: physically possible to attain and allowing the algorithm to be performant.

(c) How much information do agents need to exchange?

3. (a) Which subsystems need to be easily adaptable?

   (b) Given adaptable subsystems, how much does changing the behavior of one of them affect the others?

1.3 Hypotheses

1. Complex agents can reduce overtime and lateness by at least 10%, when compared to a greedy agent (definition of greedy below).

2. When increasing the ratio between the area an agent can observe and the area that has to be serviced, overtime and lateness decrease proportionally.

2 Multi-agent System design

2.1 Design

2.1.1 Naive greedy algorithm

We made two versions of the greedy algorithm for comparison: a version where the agents have global information and a version where they don't. We made the global version first but when we ran the experiments we noticed that it outperformed all of our solutions. To test whether this was due to the use of global knowledge we adapted the global version to only use local information.

Global The global information variant of the greedy agent works as follows. In each tick it will do the following:

- If it doesn't have a currently selected parcel it looks up the 20 closest parcels in the plane and selects the closest one that is available for pick-up.

- Then it looks at all of the parcels it has in its cargo and if parcels are deliverable it picks the one that is the closest.

- It will always prefer to deliver if it can.

- If no parcels are available yet it stands still, if all parcels have been delivered it returns to the depot.

Local The local variant behaves in a way similar to the global variant except for the following differences:

- Agents can only observe parcels within a specific radius (0.5 km).

- Agents select the most nearby parcel, if they have no deliverable cargo.

- If they do have deliverable cargo, then agents will always prefer to handle those parcels first, from nearest to farthest.
- If agents do not have deliverable cargo and cannot see any parcels, they will select the earliest parcel from their cargo to become deliverable.
- If agents cannot see any parcels and do not have any cargo, they will wander around randomly, only changeing direction when they get close to the edge.

2.1.2 Approaches taken

Some principles we applied in our solution:

1. Avoid long-term planning for agents. (To avoid cascades of schedule changes when a new package is announced.)
2. Try to spread agents evenly across the domain.
3. Avoid the need for global communication.
4. Attempt not to waste time waiting for parcels to become available.

2.1.3 Final algorithm

For our final agents we also evaluated a couple of alternatives. Specifically different strategies for assigning values to parcels and choosing which parcel to handle at a given time were implemented and different combinations were evaluated. The agents are deliberative in the sense that they will try to evaluate possible futures and pick the most valuable future to strive towards. They rely on direct inter-agent communication, through broadcasting, for knowledge about their environment. Communication is also used to coordinate: agents can bid on parcels that have been announced and use information about other agents' bids to decide whether to go pick up a parcel or not.

In general the agents behave as follows:

- At the beginning of a tick an agent looks around to check which parcels it can see, and take note whenever a parcel is missing that should be visible.
- Afterwards it assigns a value to every parcel it has seen or heard about from other agents. This happens according to an interchangeable value assignment strategy.
- It then bids on the parcels it knows about: this means that it checks whether it already has information about the value that other agents assign to the parcels and if the value it has assigned to a parcel is higher than the bids it knows about. If so it will put a message containing its own bid in a queue to be broadcasted to other agents. The broadcast radius is equal to the radius of the field of vision.

- Only one bid can be sent in each tick. A bid message also contains information about the 10 most recently "vanished" parcels. Vanished in this case means that it was previously visible to agents as being announced or available for pick-up but isn't any more. Parcels that have been picked up by the sending agent also get included in this list.

- When the bid messages have been added to the queue it selects the parcel that seems to be the best parcel to handle right now. This selection happens according to an interchangeable selection strategy. Only parcels for which it thinks it won the bid are considered. It is important to note here that it does have a form of inertia, to prevent it from changing its mind (and direction of travel) too often. This means that it is more likely to select the parcel it selected in the previous tick.

- If the selection strategy yielded a parcel to handle appropriate action is taken. If the selected parcel is in the agent's cargo it will move towards its delivery location. If it is not yet picked up the agent moves towards the pick-up location.

- If the pick-up or delivery location for the currently selected parcel has been reached and the parcel is still available it will be picked up or delivered, respectively.

- If no parcel was selected the agent will choose some direction to move in. This direction is random if no parcels or agents are present within the agent's field of vision. If, however, there are parcels or agents present these will influence the direction the agent will travel in. It will choose to move slightly toward parcels and away from agents.

- As bid messages have a limited Time To Live, they have a limited life span, but also a limited range. Agents will keep communicating about parcels that they believe are still available, but immediately cease to communicate about parcels that they know to have disappeared. In this way agents knowledge about parcels dies out. When an agent no longer encounters parcels and has forgotten about all previous parcels it will return to the depot.

EarlySelection Looking at all the bids an agent is winning, it will select the parcel whose pickup (or delivery; due to recent changes, agents unintentionally ignore their cargo in this step) time window will end the earliest, attempting to minimize lateness. If it is not currently winning any bids it will select the parcel in its cargo that has become deliverable the earliest.

BestFutureSelection An agent selects "future candidates": the parcels it is bidding the highest on (and winning) and every parcel in its cargo that it would bid higher on than the lowest bid on the former. It also selects a number of "future backers", randomly from the bids it is winning. It then proceeds to calculate the expected total amount it will bid for every future candidate if it would handle that candidate first. Finally it selects the future candidate with the highest predicted 'yield'.

TrivialValueStrategy This returns a constant (10) for any parcel. We originally used this as a simple placeholder, not intending to keep it around. However it turned out to be remarkably difficult to have the system cope with so many equal bids, which we would have never found out if we had not used this trivial implementation. In addition the performance of this strategy was unexpectedly high and we decided against removing it.

SimpleValueStrategy This strategy takes into account the distance to a parcel, when it is available for pickup/delivery and the amount of cargo an agent is currently carrying. The calculated value is equal to the inverse of the product of the amount of cargo and the time penalty multiplied by a large constant (to avoid running into the limitations of double-precision floating point numbers). This time penalty is calculated as the absolute value of the difference in time between when a parcel becomes available and when the agent would arrive at that parcel if it set out for it immediately. Hereby penalizing waiting for a parcel to become available and being late for a parcel's pickup or delivery.

2.2 Comparison with existing MAS

Gradient fields When our agents are not handling a parcel, e.g. because they aren't winning any bids, they wander around randomly, but are influenced by the presence of other visible agents which repel them and visible parcels which attract them. This resembles the gradient field approach in other MAS but is used solely to distribute agents over the entire plane, allowing for a higher concentration of agents where there are more parcels.

Auction protocols The way agents communicate is based on bidding for how much they value a parcel. This resembles an English auction protocol. However there is no auctioneer responsible for allocation of a parcel, on the contrary as soon as an agent 'hears' a higher bid it knows it has lost the bid and will communicate this higher bid to others, not bidding on that same parcel unless its perceived value becomes even higher than the currently winning bid.

3 Experiments

3.1 Setup

We have used RinSim with the Gendreau dataset to study the behaviour of our algorithms. We then ran every instance of the dataset for every configuration of interest, averaging over three repeats.

3.2 Results

In the following tables TT stands for Traveltime, L for Lateness and O for Overtime, as in gendreau et al. (2006).

	1_240_24				1_240_33				1_450_24			
	TT	L	O	Total	TT	L	O	Total	TT	L	O	Total
bestfuture simple hcommr hrui	1751	847	322	2920	2181	10174	1304	13659	7089	281	132	7503
bestfuture simple hcommr mrui	1764	895	318	2976	2160	9811	1285	13255	7040	270	104	7413
bestfuture simple hcommr lrui	1786	961	352	3099	2139	9553	1253	12945	7020	224	85	7328
bestfuture simple mcommr hrui	2087	1880	658	4625	2593	11393	1723	15709	7326	635	429	8390
bestfuture simple defaults	2087	1422	655	4164	2361	11153	1478	14993	7397	591	490	8477
bestfuture simple mcommr lrui	1945	1459	516	3920	2376	11013	1506	14895	7251	426	359	8036
bestfuture simple lcommr hrui	2263	2833	813	5909	3214	14632	2313	20159	7806	1090	838	9734
bestfuture simple lcommr mrui	2267	2232	815	5315	2984	14141	2104	19229	7690	1025	742	9457
bestfuture simple lcommr lrui	2571	2671	1120	6362	3063	13566	2177	18806	7950	1238	971	10159

Table 1: Communication Radius Variations

	1_240_24				1_240_33				1_450_24			
	TT	L	O	Total	TT	L	O	Total	TT	L	O	Total
bestfuture simple httl	2031	1662	594	4287	2517	12086	1630	16232	7289	524	319	8131
bestfuture simple defaults	2087	1422	655	4164	2361	11153	1478	14993	7397	591	490	8477
bestfuture simple lttl	2035	1543	612	4190	2555	10665	1706	14926	7462	503	603	8568
bestfuture simple lcommrel	2000	1353	582	3935	2434	10560	1571	14565	7361	531	512	8403

Table 2: Communication TTL and Reliability variation

	1_240_24				1_240_33				1_450_24			
	TT	L	O	Total	TT	L	O	Total	TT	L	O	Total
bestfuture simple defaults	2087	1422	655	4164	2361	11153	1478	14993	7397	591	490	8477
bestfuture simple hfutures	2034	1742	596	4372	2460	11228	1580	15268	7289	323	379	7990
bestfuture simple lfutures	2042	1722	613	4377	2450	10927	1560	14936	7232	456	322	8009
bestfuture simple hinertia	1957	1352	504	3813	2492	11135	1612	15239	7336	411	442	8189
bestfuture simple linertia	1982	1572	561	4115	2520	10688	1641	14849	7270	391	355	8016
bestfuture simple hpunctuality	1912	1720	516	4148	2440	10635	1566	14641	7241	510	405	8156
bestfuture simple lpunctuality	1963	1490	496	3949	2417	10880	1537	14834	7546	784	525	8855

Table 3: Best Future Parameter Variation

	1_240_24				1_240_33				1_450_24			
	TT	L	O	Total	TT	L	O	Total	TT	L	O	Total
bestfuture simple defaults	2087	1422	655	4164	2361	11153	1478	14993	7397	591	490	8477
bestfuture trivial	1912	1007	498	3417	2279	9790	1404	13474	7061	141	211	7413
early simple	1045	7008	980	9034	1283	16880	1638	19801	2408	8687	1359	12453
early trivial	1184	6152	967	8303	1328	16353	1645	19326	3270	8246	1301	12817
greedy	982	4079	816	5877	1247	16896	1694	19837	1415	3771	555	5741
greedy global	1139	307	120	1565	1236	6850	919	9005	3286	107	34	3427

Table 4: Best Future with Trivial Value Strategy, Early and Greedy Agents

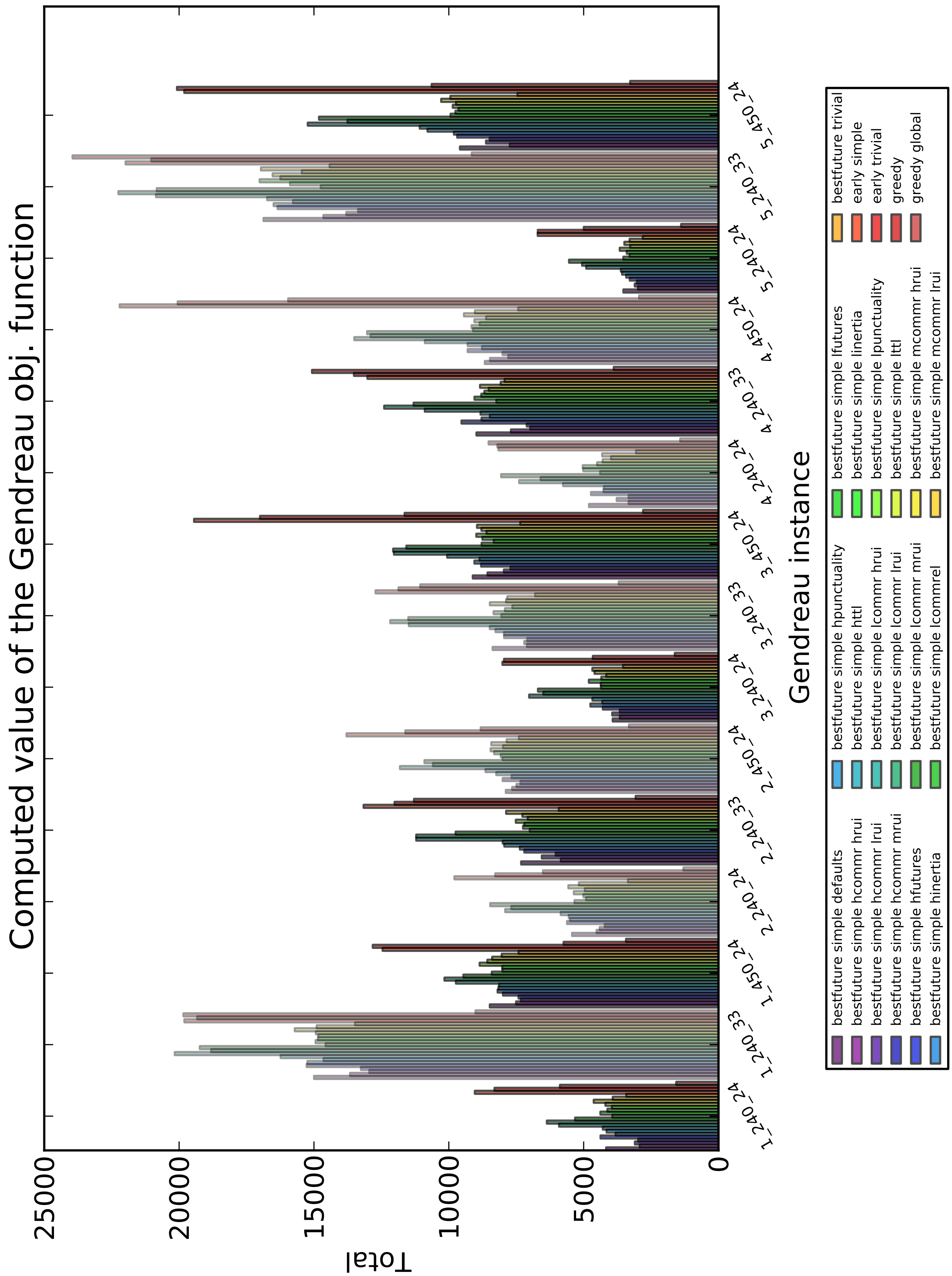


Figure 1: Overview of Total for every configuration

3.3 Analysis

The following figures focus on some interesting results.

3.3.1 General comparison between 'Greedy' and 'Complex' agents

Figures 2 and 3 show a comparison between several configurations including both greedy variants and combinations between both strategies for value assignment and parcel selection. Out of the complex agents BestFuture-Trivial has the best performance in every instance. The configurations with the Early selection strategy are generally the worst, being beaten by Greedy Local in all but 4 instances. The influence of the selection strategy is the greatest, the value assignment strategy gives approximately a fixed ratio in difference.

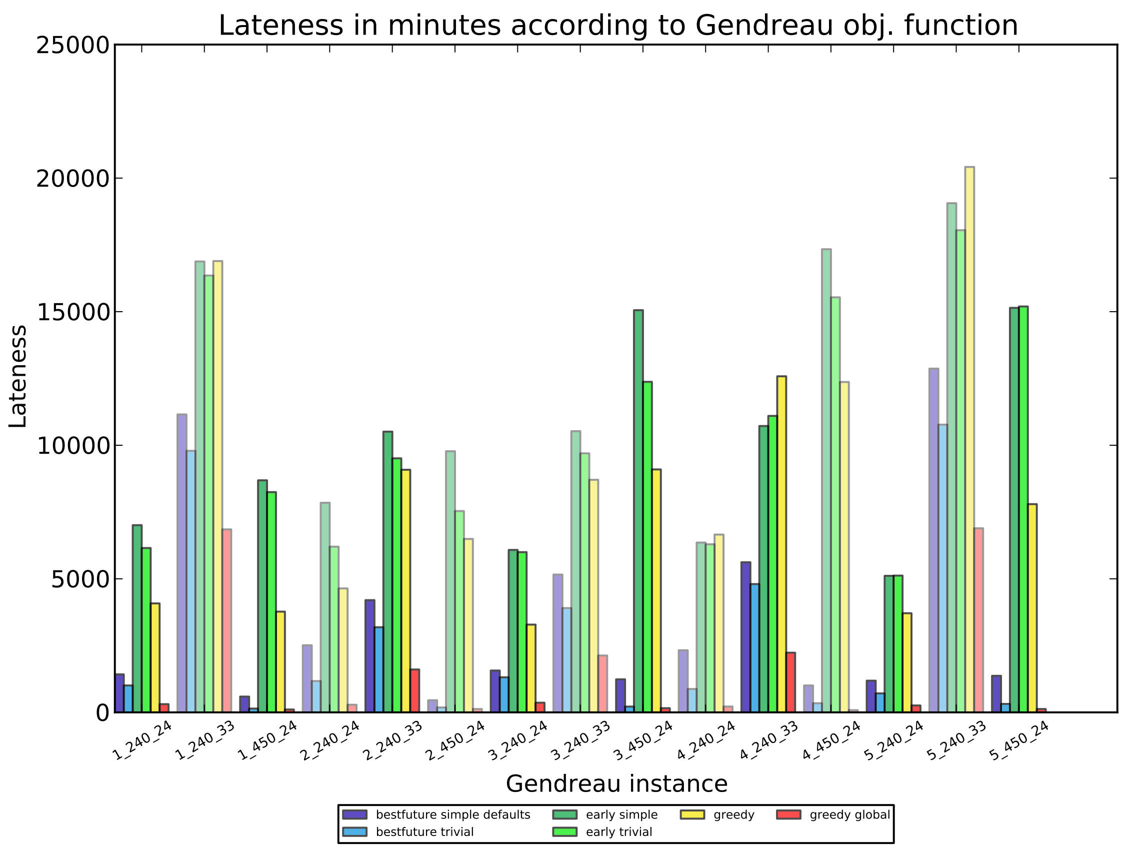


Figure 2:

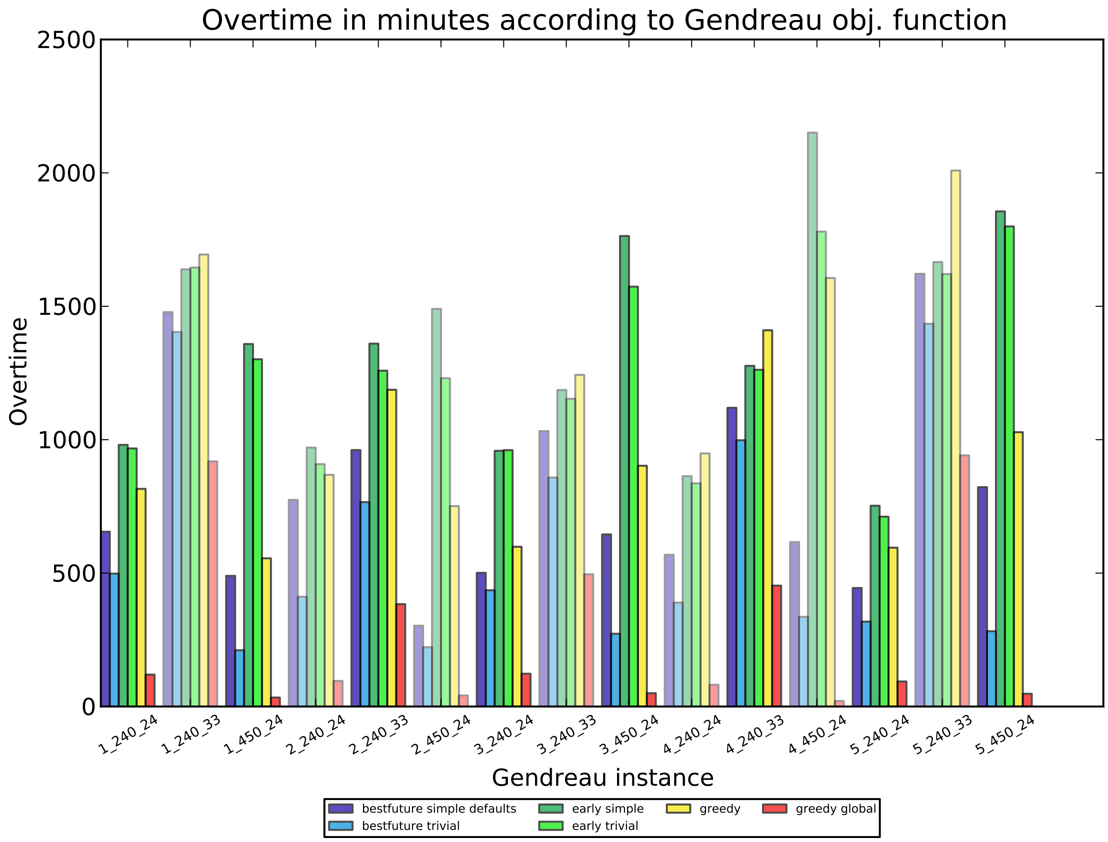


Figure 3:

3.3.2 Comparison between different strategies for the 'Complex' agent

Figures 4 and 5 show a comparison between several configurations that all have the Simple value assignment strategy and the BestFuture selection strategy for different values of parameters such as the amount of futures to consider or the punctuality, i.e. the amount of time the agent is prepared to wait for a parcel to become available and the inertia, i.e. how much better an alternative has to be to replace the currently selected parcel. It is clear from these figures that these parameters, or at least the differences in values we chose, have little effect on the performance.

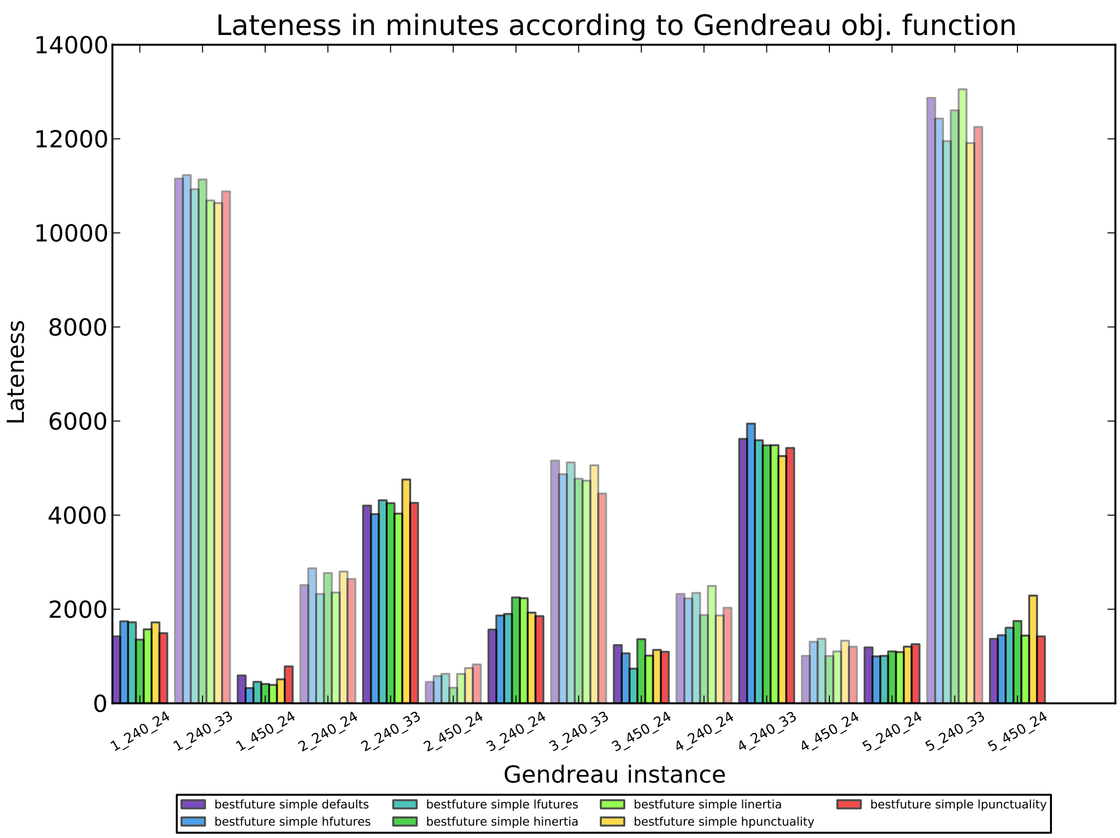
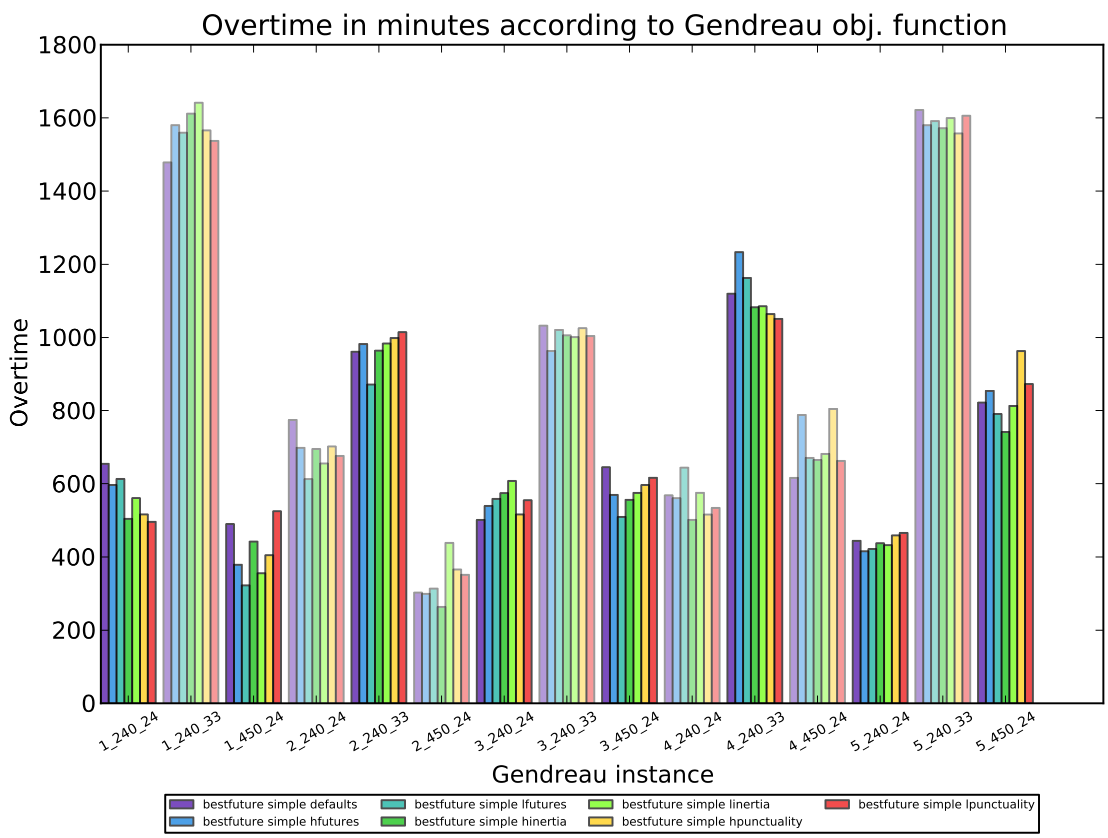

Lateness in minutes according to Gendreau obj. function
Lateness
Gendreau instance
0
2000
4000
6000
8000
10000
12000
14000
1_240_24
1_240_33
1_450_24
2_240_24
2_240_33
2_450_24
3_240_24
3_240_33
3_450_24
4_240_24
4_240_33
4_450_24
5_240_24
5_240_33
5_450_24
bestfuture simple defaults
bestfuture simple hfutures
bestfuture simple lfutures
bestfuture simple hinertia
bestfuture simple linertia
bestfuture simple hpunctuality
bestfuture simple lpunctuality

Figure 4:



Overtime in minutes according to Gendreau obj. function
Overtime
Gendreau instance
1800
1600
1400
1200
1000
800
600
400
200
0
1_240_24
1_240_33
1_450_24
2_240_24
2_240_33
2_450_24
3_240_24
3_240_33
3_450_24
4_240_24
4_240_33
4_450_24
5_240_24
5_240_33
5_450_24
bestfuture simple defaults
bestfuture simple hfutures
bestfuture simple lfutures
bestfuture simple hinertia
bestfuture simple linertia
bestfuture simple hpunctuality
bestfuture simple lpunctuality

Figure 5:

3.3.3 Influence of agent parameters on performance

Figures 6 and 7 show overtime and lateness for configurations with several values for both the communication (and sight) radius and the influence parcels and other agents have on the agents' random movements. The figures show a clear linear dependence of lateness on the communication radius and don't seem to indicate a dependence on the road user influence. The overtime gets significantly worse as the communication reliability decreases. Figures 8 and 9 show overtime and lateness for configurations with several values for the bid messages' Time To Live. A high time to live does seem to decrease performance in a couple of instances, especially with regards to lateness.

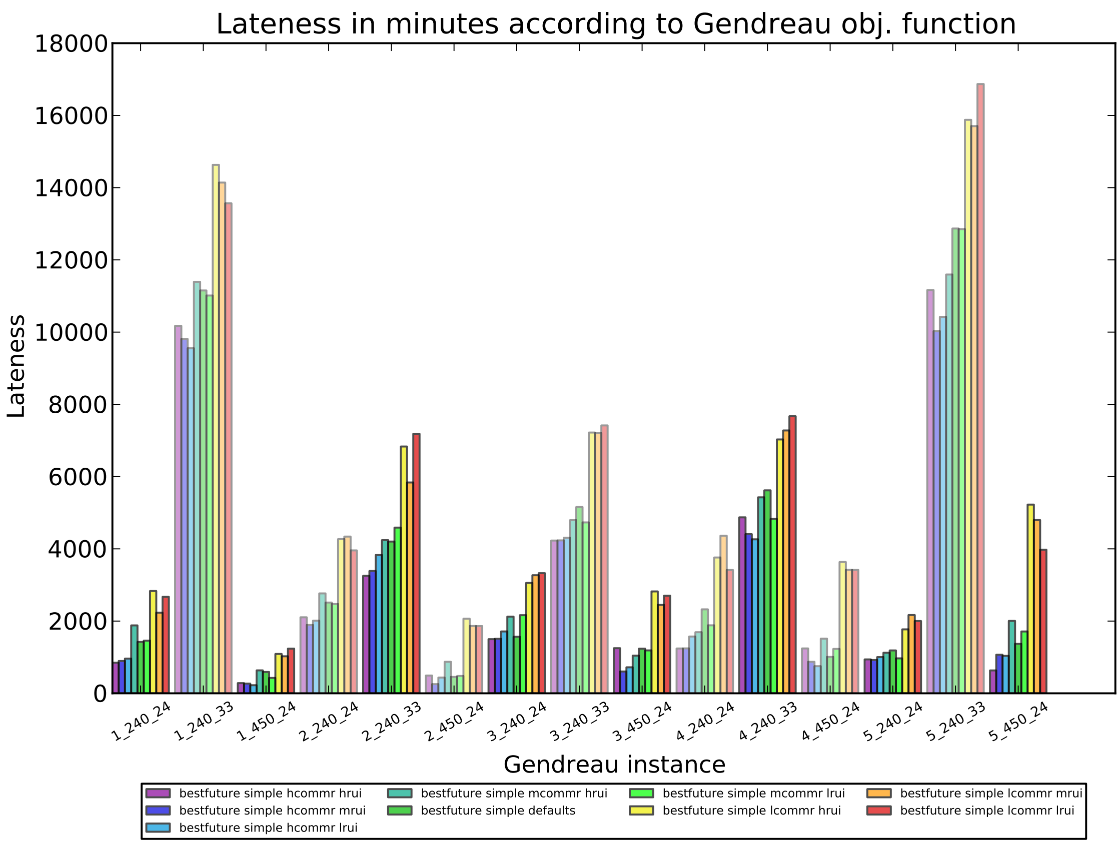


Figure 6:

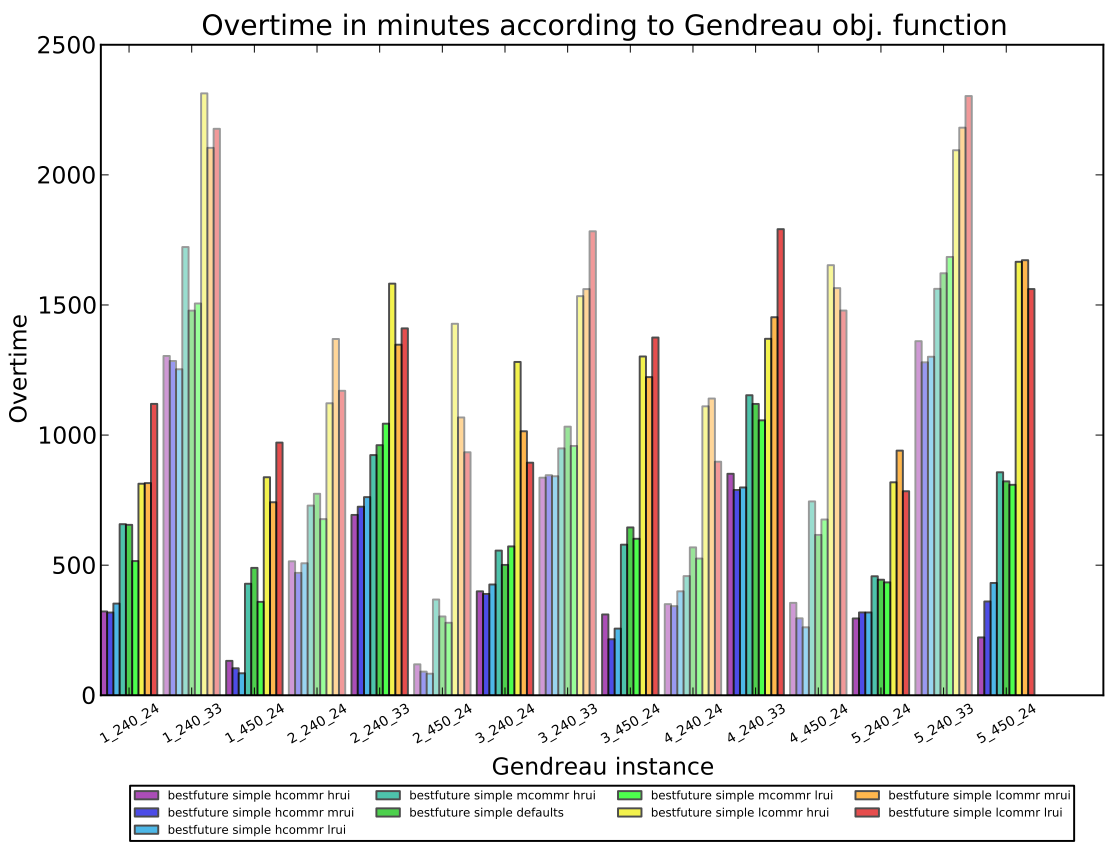

Overtime in minutes according to Gendreau obj. function
Overtime
Gendreau instance
0
500
1000
1500
2000
2500
1_240_24
1_240_33
1_450_24
2_240_24
2_240_33
2_450_24
3_240_24
3_240_33
3_450_24
4_240_24
4_240_33
4_450_24
5_240_24
5_240_33
5_450_24
bestfuture simple hcommr hrui
bestfuture simple hcommr mrui
bestfuture simple hcommr lrui
bestfuture simple mcommr hrui
bestfuture simple defaults
bestfuture simple mcommr lrui
bestfuture simple lcommr hrui
bestfuture simple lcommr mrui
bestfuture simple lcommr lrui

Figure 7:

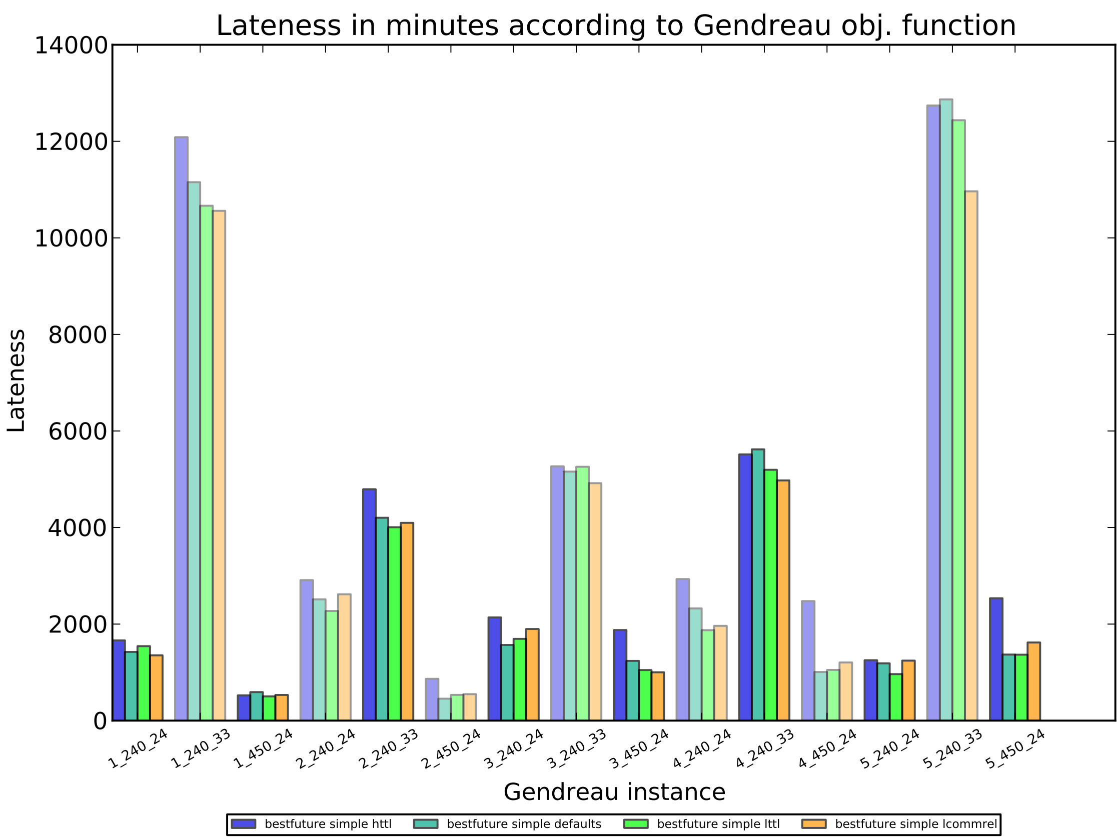

Lateness in minutes according to Gendreau obj. function
14000
12000
10000
8000
6000
4000
2000
0
Lateness
1_240_24
1_240_33
1_450_24
2_240_24
2_240_33
2_450_24
3_240_24
3_240_33
3_450_24
4_240_24
4_240_33
4_450_24
5_240_24
5_240_33
5_450_24
Gendreau instance
bestfuture simple httl
bestfuture simple defaults
bestfuture simple lttl
bestfuture simple lcommrel

Figure 8:

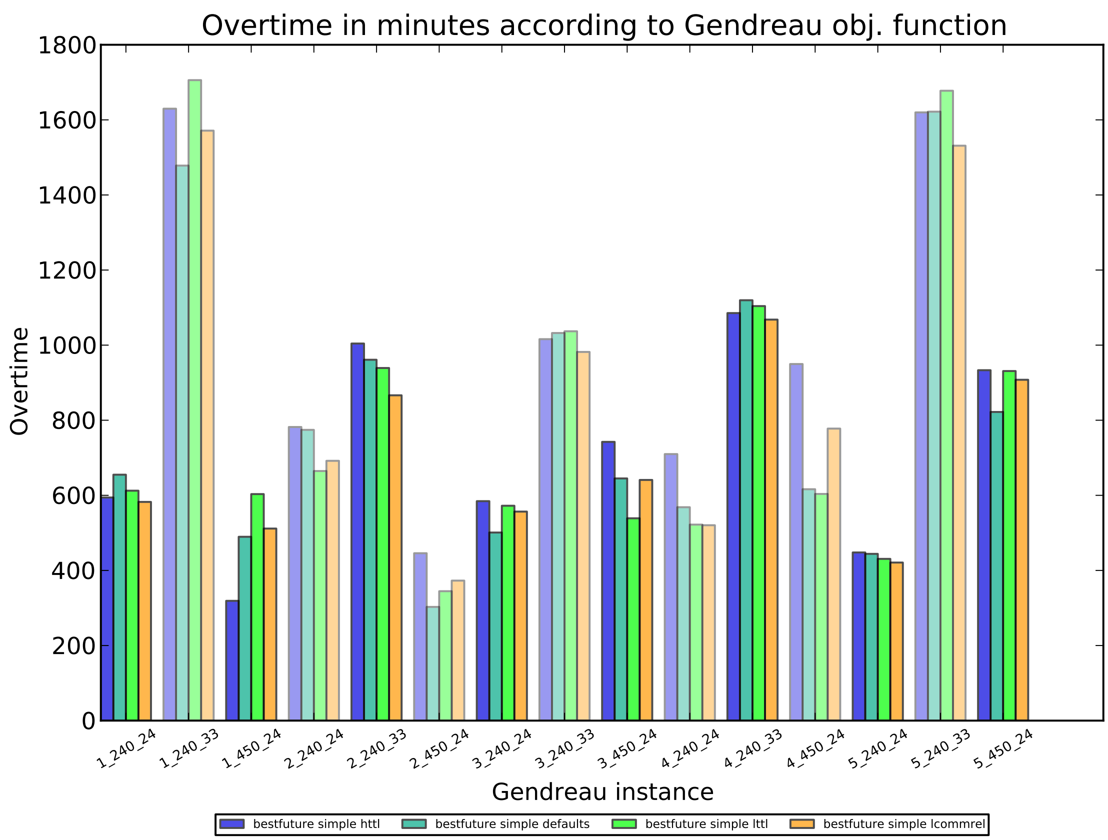

Overtime in minutes according to Gendreau obj. function
Overtime
0
200
400
600
800
1000
1200
1400
1600
1800
1_240_24
1_240_33
1_450_24
2_240_24
2_240_33
2_450_24
3_240_24
3_240_33
3_450_24
4_240_24
4_240_33
4_450_24
5_240_24
5_240_33
5_450_24
Gendreau instance
bestfuture simple httl
bestfuture simple defaults
bestfuture simple lttl
bestfuture simple lcommrel

Figure 9:

4 Conclusion

4.1 Objectives

Reduced lateness and overtime Compared to the global knowledge greedy agent our final algorithm doesn't perform very well. However, compared to the local knowledge greedy agent, in all tests except one, all of the configurations for our final algorithm that use the BestFuture selection strategy and have at least the communication radius we designated as medium outperform the local knowledge greedy agent.

Feasibility Our final algorithm does not rely on global knowledge in any way. Communication with the radius and reliability that we assume in our algorithm seems to be possible in real applications: for example in this Kickstarter project.

Adaptability Different strategies for value assignment and parcel selection have been implemented already. Though at the moment we have only experimented with setups where every agent has the same strategies and parameters there is no reason why more heterogeneous setups wouldn't work. We claim that our system can be adapted to work in a high number of significantly different environments just as well as in our experiments.

4.2 Questions

- Q How does the frequency of parcel announcements influence lateness and overtime?

  A The results for the high frequency data sets are worse for our algorithm, which can be seen when comparing our statistics to those shown in the original Gendreau et al. (2006) paper. This may be due to the fact that our agents generally resist having too many items in their cargo at once and therefore may miss opportunities to pick up a couple of conveniently placed parcels in one go. However one of our configurations, BestFuture-Trivial, does have this tendency and performs rather well in these scenarios.

- Q How does the number of agents influence lateness and overtime?

  A The results for the datasets with 20 agents indicate that having more agents provides a significant advantage in our case. This is probably because of the added possibilities for communication and the larger total coverage of the pick-up area,

- Q Can more complex behaviour, i.e. taking into account more variables, decrease lateness and overtime?

  A It seems that in our case the selection strategy is the most important area where performance can be gained. Taking more variables into account for the value function even decreased performance.

- Q How does performance depend on the amount of knowledge an agent is able to acquire about his surroundings?

  A The experiments where we chose different values for the communication radius (and so the line of sight) show that this is an important factor. Agents with a larger field of vision performed consistently better.

- Q What is a reasonable communication range between agents?

  A From what we've seen reducing the communication range significantly reduces the performance so it should probably be set as high as technically feasible, further experiments might reveal that setting it too high reduces the increase in performance which would make it more economically interesting to tradeoff range for cost.

- Q How much information do agents need to exchange?

  A We tried to avoid sending information about parcels that were picked up in the beginning but this quickly turned out to be very detrimental to performance. What ended up happening was that agents kept driving around looking for parcels that were supposed to be there according to their memory but weren't, even though all of the parcels were already picked up and delivered. To keep the amount of data that needs to be transferred low we limited the amound of unavailable parcels communicated in each message.

- Q Which subsystems need to be easily adaptable?

  A We chose the value assignment strategy and parcel selection strategy, but most likely the subsystems for interacting with parcels (finding them, picking them up, ...) and communicating (with other agents, but possibly also with the depot or customers in other set-ups) should be made more adaptable as well. However for the purpose of our simulations this was not necessary.

- Q Given adaptable subsystems, how much does changing the behavior of one of them affect the others?

  A This was hard to measure in our case as the influence of the Early selection strategy was so great it overpowered the influence of the alternatives for the value assignment strategy.

4.3 Hypotheses

1. • Complex agents can reduce overtime and lateness by at least 10%, when compared to a greedy agent.

   > As long as the greedy agent can only gather as much information about its environment as the complex agent this appears to be true according to our experiments.

2. • When increasing the ratio between the area an agent can observe and the area that has to be serviced, overtime and lateness decrease proportionally.

   > In some of the scenarios this seems to be the case but certainly not in all.

4.4 Critical reflection

We were honestly pretty surprised that the Trivial value assignment strategy outperformed the Simple value assignment in so many cases. When looking at figure 1 you can see a dip right before the peak of the 'early' methods. In hindsight bestfuture trivial and greedy global have very similar behaviour, while the first approximates the global information through communication, the second does not attempt to minimize the distance it travels to the next parcel.

At first we planned implementing a third ValueStrategy, a more complex function taking into account the (or a near) optimal path along all parcels for which an agent was winning the bids. After experimenting with the simple value function and considering the difficulty of finding an 'optimal' path we decided against this. For the simple value function we tried different exponents of the considered factors and also addition and weighting instead of multiplication, all of these performed even worse.