Urban Traffic Control with Co-Fields

Marco Camurri, Marco Mamei, and Franco Zambonelli

Dipartimento di Scienze e Metodi dell'Ingegneria,
University of Modena and Reggio Emilia
Via Allegri 13, 42100 Reggio Emilia, Italy
{camurri.marco.31767, mamei.marco, franco.zambonelli}@unimore.it

Abstract. Traffic control can be regarded as a multiagent application in which car-agents and traffic-light-agents need to coordinate with each other to optimize the traffic flow and to avoid congestions. Environment abstractions naturally suit this scenario in that agents actions are mainly driven by traffic-related information that are distributed across the environment both at a practical and conceptual level. In this context we present traffic-control mechanisms on the basis of our Co-Fields model and discuss some experimental results we obtained in simulations that validate our proposal.

1 Introduction

Traffic management is a very complex problem. Millions of people every day suffer from congestion in urban road networks. This has led researchers to wonder whether is possible to regulate traffic flow in order to reduce the congestion (i.e., to reduce the average time that a car takes to reach its destination) [1–4, ?]. Apart from long-term structural solutions (i.e., building new roads), it is possible in principle to control the urban traffic flow either acting on the individual cars (that could try to move avoiding traffic jams) or acting on the traffic-lights (that could try to dynamically optimize the overall throughput).

Both these candidate approaches naturally suit the agent paradigm and it is rather easy to conceive such a scenario in terms of car-agents and traffic-light-agents coordinating their actions to improve the traffic flow. It is also rather easy to understand that these kind of applications (where the agents' goal is inherently related to their motion in an environment) are killer-applications with regard to the environment-in-multi-agent-system idea. In this context, in fact, environment abstractions are natural and their power in decoupling agents' actions and in providing context-awareness is undisputable [5, 6].

The main contribution of this paper is to illustrate how the Co-Fields [7] model can be fruitfully applied to this application scenario, and to present some experimental results we obtained (in simulations) adopting it.

The Co-Fields model is based on distributed data structures spread across an environment implementing digital mockups of physical fields (such as the gravitational one). Such distributed data structures mediate and rule the coordination activities among agents that act on the the basis of the perceived local field configuration [7]. In our case study, a *traffic* field will be used to convey aggregated information about the current traffic conditions. In each area, this field will have a magnitude proportional to the amount of traffic in there. Car-agents can use such information (which is actually an environment abstraction) to coordinate their motion avoiding traffic jams. In a similar way, traffic-light-agents can tune their green-light policy according to the perceived value of the *traffic* field.

The rest of this paper is organized as follows. Section 2 presents some backgrounds. In particular, it briefly illustrates the Co-Fields model and describes the GLD (Greed Light District) simulation platform where we run our experiments. Section 3 presents our proposal to control traffic on the basis of Co-Fields. Section 4 shows and discusses experimental results. Finally, Section 6 concludes and sketches future works.

2 Backgrounds

In this section we will briefly review the coordination model Co-Fields [7] that is at the heart of our proposal, and we present the Green Light District [8] platform used to perform simulations and experiments on realistic traffic situations.

2.1 Co-Fields

Co-Fields is a coordination model for multiagent systems in which environment abstractions have a central role. The main idea in Co-Fields is to provide agents with an effective and easy-to-use representation of their operational environment. To this end, Co-Fields delegates to the infrastructure the task of constructing and automatically updating an essential distributed "view" of the system situation - possibly tailored to application-specific coordination problems - that "tells" agents what to do (i.e., how to act to implement a specific coordination patterns). Agents are simply let with the decision of whether to follow such a suggestion or not.

To achieve this goal, we take inspiration from the physical world, i.e., from the way particles in our universe move and globally self-organize accordingly to that contextual information which is represented by potential fields. In particular, in our approach, contextual information is expressed in the form of distributed computational fields (Co-Fields). A computational field is a distributed data structure characterized by a unique identifier, a location-dependent numeric value, and a propagation rule identifying how the field should distribute across the environment and how its value should change during the distribution. Fields are locally accessible by agents depending on their location, providing them a local perspective of the global situation of the system. Each agent of the system can generate and have propagated specific fields across the environment, conveying application-specific information about the local environment and/or about itself. Agents can locally perceive these fields and act accordingly, e.g. following the fields' gradient. The result is a globally coordinated behavior, achieved with very little efforts by agents.

More in detail, the Co-Fields approach is centered on a few key concepts:

1. Contextual information is represented by "computational fields", spread by agents and/or by the infrastructure, diffused across the environment, and locally sensed by agents;
2. A coordination policy is realized by letting the agents act on the basis the local field configuration, the same as a physical mass moves in accord to the locally sensed gravitational field.
3. Both environment dynamics and agents' actions may induce changes in the fields' surface, automatically propagated by the infrastructure, thus inducing a feedback cycle (point 2) that can be exploited to globally achieve a global and adaptive coordination pattern.

In the next section we will present how these model can be easily applied to the traffic control problem.

2.2 Green Light District

GLD (Green Light District) is a program that performs discrete simulations of road networks. The full application consists of two parts: an Editor and a Simulator. The Editor enables the user to create an infrastructure (a road map) and save it to disk. The Simulator can then load the map and run a simulation based on that map. Before starting a simulation, the user can choose which traffic light controller and which driving policy will be used during the simulation. A **traffic light controller** (or simply TL-Controller) is an algorithm that specifies the way traffic lights are set during the simulation (i.e., it specifies traffic-lights green-red policy). A **driving policy** specifies which are the paths followed by cars to reach their destinations. Since GLD is open source, it can be easily extended in order to add new algorithms for TL-Controllers and for driving policies. This is exactly what we did to test our algorithms. In order to get a more realistic simulation, two values can be adjusted for each edge node in the map (i.e., for each node that can put cars into the map):

Spawn frequency: is the frequency (or probability) at which a node spawns new road users. Its values range between 0 and 1. For example, a spawn frequency of 0.5 for a node means that the node will spawn one car every two time steps (or cycles).

Destination frequency: a random destination point is assigned to every car entering the map. With this parameter it is possible to specify the probability distribution from which the destinations are drawn. This allows to create crowded destinations and a not-uniform traffic-flow.

While running a simulation (see Figure 1), GLD can track different types of statistics such as the number of road users that reached their destination, the average junction waiting time ore the average trip waiting time. The data collected from the simulation can be displayed in a window, or exported in text format for further analysis.

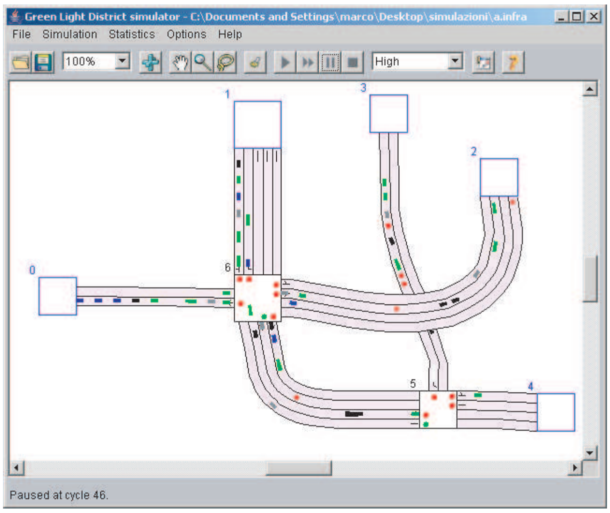


Fig. 1. The GLD traffic simulator

3 Traffic Control

In this section, we describe the Co-Fields policies we realized to implement traffic control. We realized both mechanism to control cars and traffic-lights. The effectiveness of this policies will be tested with the aid of a simulation tool.

The most direct approach to try to manage traffic is to improve the traffic light control algorithms (i.e., the way traffic lights are switched to regulate the flow of cars). So far traditional methods have tried to optimize the flow of vehicles for a given car density by setting appropriate phases and periods of traffic lights [4]. However, the use of static timing patterns does not take into account the actual state of traffic (for instance, the current speed of cars might be greater than the expected one). In other words, traditional traffic lights controller cannot adapt to the current car density, and thus they cannot manage correctly unusual and extraordinary situations (for instance, a stream of cars leaving a stadium or a big concert) [9, 10].

As alternative to the previous approach, we can imagine a network of "intelligent" traffic lights that can perceive the current state of traffic and dynamically self-organize, without the need of any complex central controller. The core idea of our approach is to induce a cooperative behavior between traffic lights, by means of simple, field-mediated interactions.

Improving traffic lights control algorithms is not the only way to make traffic management more efficient. Thanks to the wide diffusion of low-cost wireless communication devices, new approaches to the problem are becoming feasible. It is not difficult to imagine that, in a few years, each car will be equipped with a mobile computer capable of communicating with a computer network embedded in the city streets and junctions. In this way, a car could get useful information about traffic jam or queues, and suggest alternative paths to its driver. In this scenario, both traffic lights and cars can be seen as simple software agents able to interact with each other in order to get a more complex goal [11, **?**,**?**].

3.1 Co-Fields Cars

The aim of our Co-Fields driving policy is to guide the cars towards their destinations avoiding the most crowded areas. With reference to Figure 2: let's suppose that five cars are leaving node A to reach node H. If a shortest-path driving policy is used, all the cars will follow the path ACFH. If our Co-Fields driving policy is used, the last two cars will follow an alternative (possibly longer) path, because the shortest one is too crowded.

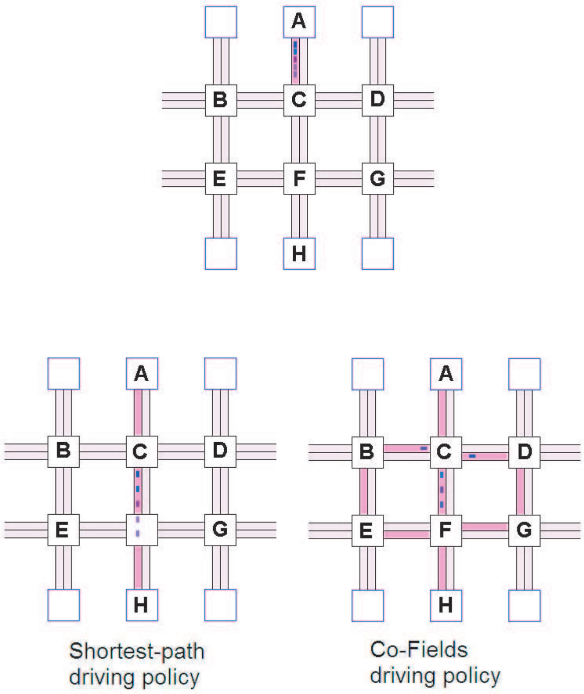
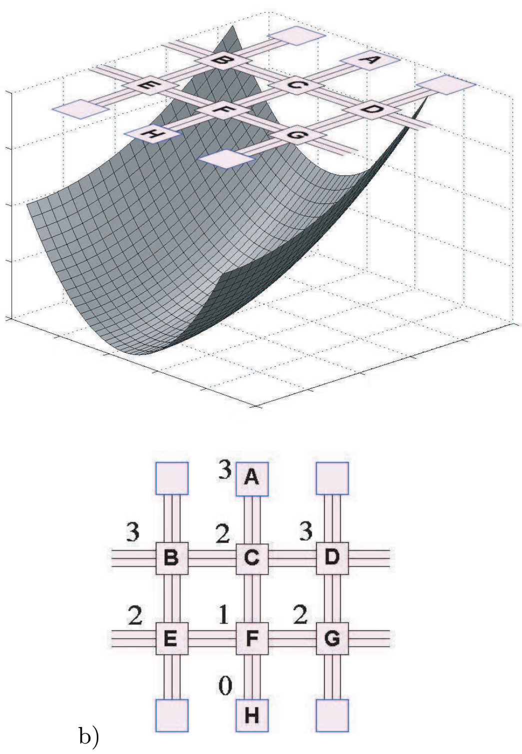


Fig. 2. Cars avoiding crowded paths

I order to obtain such policy using the Co-Field model, we need the cars to perceive two kind of fields: a distance field and a traffic field.

Distance Field

This field is generated by every node of the map, and its values does not change over time. It has value 0 in the node that generates it. In a generic node N it has a value equal to the length of the shortest path between N and the node from which the field generates. If DF_H is the distance field generated by node H, then on each node a car can perceive the fields DF_1, DF_2, ... , DF_N, where N is the number of nodes in the map.



a)

b)

Fig. 3. (a) Distance field generated by node H. (b) Values of the distance field generated by node H

Traffic field

The traffic field (TRF) represents the intensity of traffic in a given point of the map. In each lane, it has a values equals to the number of cars on the lane, divided by the length of the lane. The traffic field is dynamic and its values change over time depending on the car's movements.

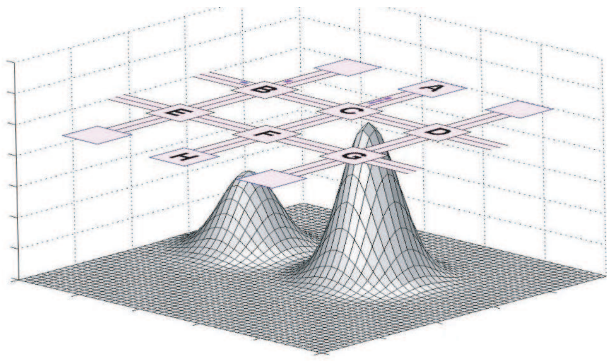


Fig. 4. Traffic field

A car heading for node H evaluates a combined field (CF) [1] as the linear combination between the distance field (DF) generated by node H and the perceived traffic field (TRF):

$$CF = DF_H + \alpha \cdot TRF$$

The firs term of the coordination field is a field that has its minimum point in correspondence of the destination node H. Because of the shape of this field, then, cars following downhill this field are guided towards theirs destinations. The second term takes into account traffic. Having its maximum points where the car density is high, the fields TRF (with $\alpha > 0$) tends to repulse cars from these points. For this reason, can be regarded as the weight assigned to traffic information in the choice of the path. If α is very small, cars will follow their shortest path without considering traffic conditions. If α is very high, cars will prefer a possibly longer path whenever the shortest one is a bit crowded (see Figure 5.

3.2 Co-Fields Traffic Lights

The key idea in our Co-Fields TL-Controller is to propagate a computational field throughout the map, that helps traffic lights to coordinate with its immediate neighbors.

Each traffic light evaluates the local value of a specific *Green Field* (GF) as the sum between the number of cars waiting in its queue and the number of cars waiting in the queues of those traffic lights that can send cars toward it. If $N(s_i)$ is the number of cars waiting for traffic light s_i, and s_1, s_2, ... s_n are the traffic lights that can send cars to s_0, then s_0 evaluates its green field as follows:

[1] This is actually the Coordination Field described in [7]

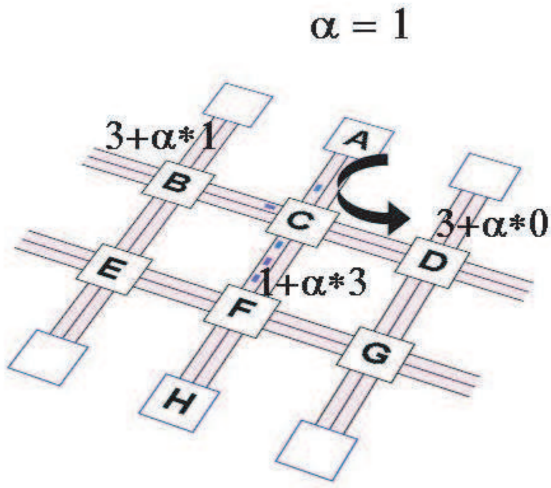


Fig. 5. A car following downhill the combined field. For example in the road between C and F there are 3 cars. The field value there is $1 + \alpha \cdot 3$

$$GF(S_0) = \frac{N(s_0)}{l_0} + \gamma \cdot (\frac{N(s_1)}{l_1} + \frac{N(s_2)}{l_2} + ... + \frac{N(s_n)}{l_n})$$

where l_i is the length of the lane regulated by s_i. Figure 6 shows how $GF(s_0)$ is computed (normalization to the length of the lanes was omitted). In the previous formula, γ measures the importance given to the traffic conditions of neighboring traffic lights. If $\gamma = 0$ there is no coordination between neighboring traffic lights: each one takes its own decision simply looking at the number of cars on its lane.

We can regard the intensity of the Green Field as a measure of the advantage that can be obtained if the underling traffic light is set to green. At every step of the simulation and for every junction, the traffic lights having the highest value of green field are set to "green", while the others are set to "red". Serious problems arise when the green field changes its values too rapidly over time, because in this case traffic lights switches from "red" to "green" very frequently, and cars spend too much time waiting for the "yellow" light to become "green". For this reason we limited the maximum rapidity of variation of GF by adding a new parameter, T_{min}, which represents the minimum time between two consecutive variations of GF.

It is worth noting that this approach is particulary interesting in that – in contrast with the vast majority of field-based applications [12] – it involves a coordination task that is not related to motion coordination.

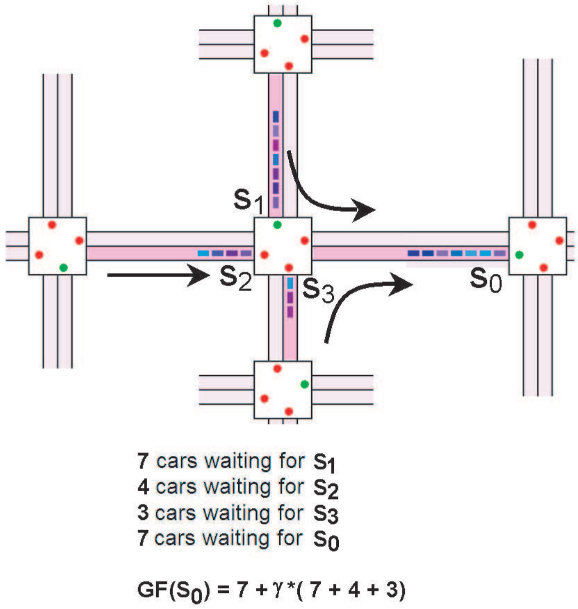


Fig. 6. Example of calculus of GF for traffic lights S0

4 Experiments

In this section we want to test that the coordination policies presented in the previous section for cars and traffic lights can improve traffic management by avoiding or reducing traffic congestion. Road networks congestion happens when the rate at which new cars enter the map is greater than the rate at which they exit. Looking at the chart that plot the number of arrived road users over time, road-net congestion appears as a flattening of the curve (it means that no new cars get to their destination).

4.1 Simulation Set-Up

For our experiments, we decided to use a map like the one in Figure 7, which consists in a square 8x8 grid of junctions. All the roads have exactly 2 lanes -one for each direction- and a length of 50 blocks. Each car has a length of 2 blocks and a speed of 2 blocks per time step. Cars enter the map from 32 edge nodes placed along the perimeter. Destination for cars spawned by north nodes is randomly chosen from south nodes; destination for cars spawned by west nodes is chosen from east nodes, and so on. This is necessary if we want the traffic to be evenly spread throughout the map.

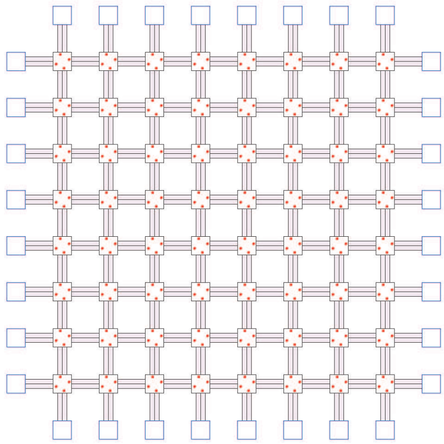


Fig. 7. Map used in the experiments

4.2 Results of the Experiments

Co-Fields traffic lights controller In order to test the performance of our Co-Fields TL-Controller, we compared it with a standard TL-Controller that simply switches lights according to a pre-determined cycle. Thus, if the standard TL-Controller is used, traffic lights remain green for T cycles of simulation, switch to red for the next 3T cycles, and then become green again, and so on. Parameter T is said to be the period of the standard TL-controller. Graphs in Figure 8 show the results of simulations on the map presented at the beginning of this section. A spawn frequency of 0.2 cars/cycle was set for every edge node. It can be seen that the Co-Fields method performs better then the standard method. To make sure that the advantage is not due to the particular choice if T and T_{min}, we set the same value for T and T_{min} (results for T = T_{min} = 4 are shown in plots B and C). Then we repeated the experiment with different values of T and T_{min} (results for T = T_{min} = 10 are shown in plots D and E). In all tests, standard TL-Controllers reach deadlock after about 2500 simulation cycles, while Co-Fields TL-Controllers manage to avoid it. The A line shows the performance of a random TL-Controller which randomly switches lights at every cycle.

Co-Fields driving policy Before comparing the performance of our Co-Fields driving policy with a standard shortest-path driving policy, we performed some simulations to find the optimal value for parameter α . During all tests, a standard TL-Controller was used (with period T = 10 cycles) and a spawn frequency of 0.2 was set for each edge node. Results obtained for various values of α (see Figure 9) indicate that the algorithm performs better when α ranges between 2 and 7.

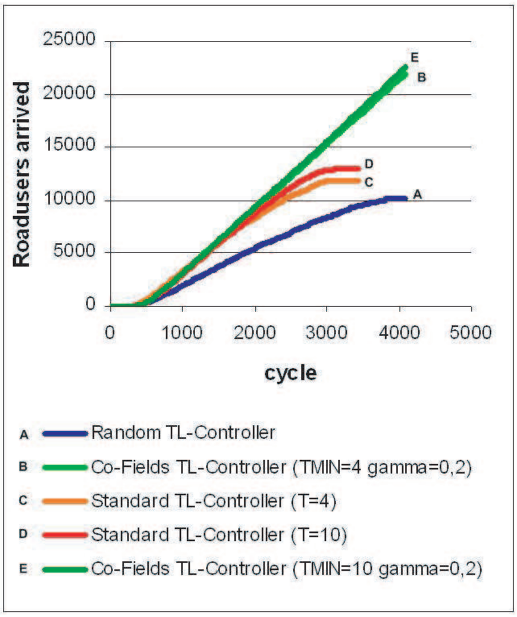


Fig. 8. Performance of various traffic light controllers

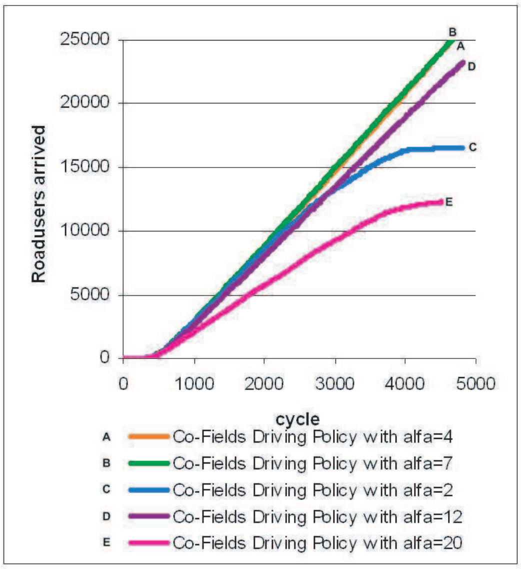


Fig. 9. Performance of Co-Fields driving policies varying α

Then we compared our Co-Fields driving policy (with $\alpha = 7$) with the shortest path driving policy (i.e., a driving policy that make cars strictly follow the shortest path towards their destinations without avoiding crowded areas). Results displayed in Figure 10 shows that the shortest path driving policy lead to deadlock after about 2500 cycles, while our driving policy avoid it.

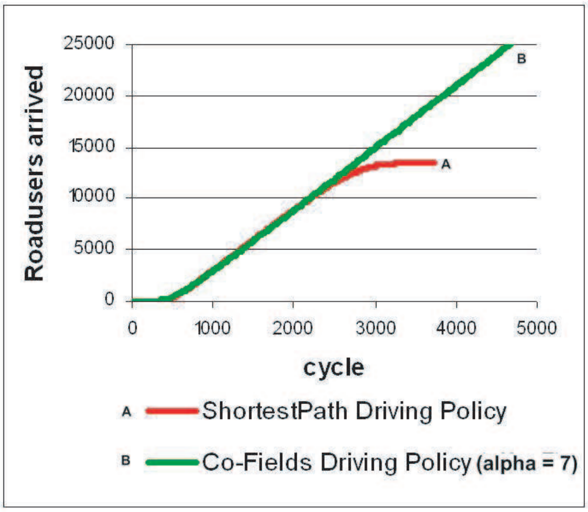


Fig. 10. Comparison between shortest path and Co-Fields driving policies

Comparison between Co-Fields cars and Co-Fields traffic lights We conclude this section with a comparison between the two presented approaches. In order to understand the interaction between the two methods, we set a very high traffic level in the map (0.3 cars/cycle for each edge node) so that both methods, separately, do not manage to avoid net congestion. Then we applied them simultaneously. Results are shown in Figure 11. The worst performance (A) is obtained when traditional methods are used both for traffic lights control and for driving policy (i.e. standard TL-Controller and shortest path driving policy). This lead to deadlock after about 1500 cycles. Better results are obtained if the Co-Fields TL-Controller or the Co-Fields driving policy are used. In this cases (B and C lines) net congestion is clearly delayed. Concurrent use of both approaches (E) yields the best performance, and manage to avoid congestion. The following parameters were used during simulations:

- Standard TL-Controller with T = 10
- Co-Fields TL-Controller with $\gamma = 0,2$ and $T_{min} = 10$
- Co-Fields driving policy with $\alpha = 4$

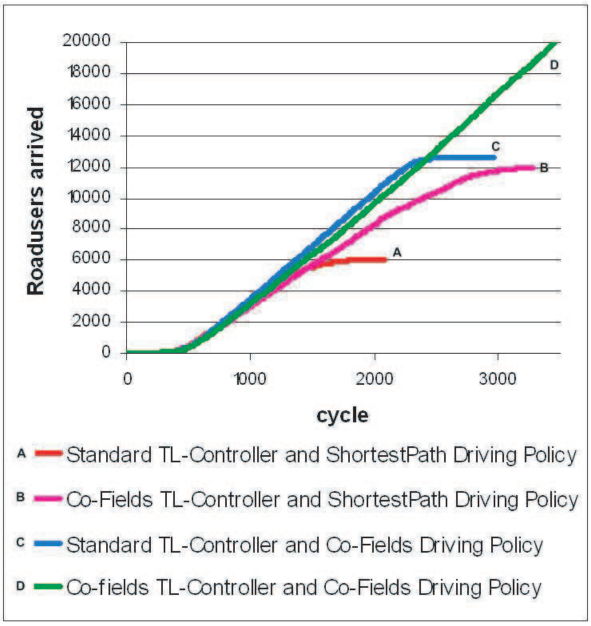


Fig. 11. Comparison between Co-Fields driving policy and Co-Fields TL-Controller

Apart from these promising experimental results and the Co-Fields strengths in managing agents' coordination activities, a correct evaluation of the Co-Fields model cannot overlook the following drawbacks:

1. Co-Fields and the strictly local perspective in which agents act promote a strictly greedy approach in their coordinated actions. In fact, agents act on the basis of their local viewpoint only, disregarding that a small sacrifice now can possibly lead to greater advantages in the future. With regard to traffic, this implies, for example, that cars would be better queue for a short-time other than looking for uncrowded path that can be longer or also (more) crowded in other areas.
2. The Co-Fields model is not supported by a well-specified engineering methodology. In other words, we still have not identified a principled way to help us identify, given a specific coordination pattern to be enforced, which fields have to be defined, how they should be propagated, and how they should be combined by the recipient agents. For example, although the presented coordination tasks are rather natural and easy to be identified, the coefficients specifying how the different fields have to be combined have to be hand-tuned.
3. From an implementation point of view, the Co-Fields approach requires a distributed computer infrastructure to store the distributed data structures

representing the fields. In the presented approach, all the possible destinations have to propagate a field across this infrastructure and this of course might create scalability concerns. However, it is rather easy to extend the current model to aggregate fields and save resources. For example, the fields associated to close destinations could be merged together to save bandwidth and storage, once a suitable distance from their sources have been reached. Such an extension, is in our future work.

5 Conclusion and Future Work

In this paper we presented a field-based approach to traffic control. Both mechanisms to control cars and traffic-lights have been presented.

While the reported tests are encouraging, a more thorough evaluation is required to better assess real world potential. In particular, we think that the following points are critical to better assess our model:

1. One obvious and important step is to use a more sophisticated simulator e.g., CORSIM [13], TRANSYT7F [14], or SimTraffic [15] that better approximates realistic traffic behavior. With a more sophisticated simulator, we could also test real intersection configurations and historical traffic patterns, which would allow a more objective assessment of the results.
2. It would be also important to measure other parameters in order to better understand the pros and cons of our proposal. Parameters such as: the average journey time, the waiting time, the number of detours cars have to take are important aspect to assess the model.
3. In addition, it would be interesting to compare our approach with more advanced and sophisticated control mechanisms both in terms of cars and traffic-lights.

Finally, a more pragmatic but nonetheless critical followup to this research would be an assessment of the costs and additional factors necessary to deploy a system like ours in an actual urban environment.

6 Acknowledgments

Work supported by the project CASCADAS (IST-027807) funded by the FET Program of the European Commission.

References

1. Fehon, K., Chong, R., Black, J.: Adaptive traffic signal system for cupertino california. In: Asia-Pacific Transportation Development Conference, San Francisco (CA), USA (2003)
2. Ferreira, E., Subrahmanian, E., Manstetten, D.: Intelligent agents in decentralized traffic control. In: IEEE Intelligent Transportation Systems. IEEE CS PRess (2001)

3. Gershenson, C.: Self-organizing traffic lights. Complex Systems **16** (2005) 29 – 53
4. Wiering, M.: Intelligent traffic light control. ERCIM News, European Research Consortium for Informatics and Mathematics **53** (2003) 40 – 41
5. Weyns, D., Parunak, V., Michel, F., Holvoet, T., Ferber, J.: Environments for Multiagent Systems, State-of-the-art and Research Challenges. Springer Verlag - LNAI 3374 (2005)
6. Weyns, D., Parunak, V., Michel, F., Holvoet, T., Ferber, J.: Environments for Multiagent Systems, State-of-the-art and Research Challenges. Springer Verlag - LNAI 3830 (2006)
7. Mamei, M., Zambonelli, F., Leonardi, L.: Co-fields: A physically inspired approach to distributed motion coordination. IEEE Pervasive Computing **3** (2004) 52 – 61
8. (Green-Light-District) http://www.students.cs.uu.nl/swp/2001/isg/.
9. Dresner, K., Stone, P.: Multiagent traffic management: a reservation-based intersection control mechanism. In: International Joint Conference on Autonomous Agents and Multiagent Systems. ACM Press, Melbourne, Australia (2004)
10. Nigarnjanagool, S.: Development and evaluation of an agentbased adaptive traffic signal control system. In: Conference of Australian Institutes of Transport Research. (2003)
11. Bandini, S., Bogni, D., Manzoni, S., Mosca, A.: St-modal logic to correlate traffic alarms on italian highways: Project overview and example installations. In: International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems. Volume 3533 of Lecture Notes in Computer Science. Springer (2005)
12. Mamei, M., Zambonelli, F.: Field-based Coordination for Pervasive Multiagent Systems. Springer Verlag (2005)
13. (Federal-Highway-Administration-CORSIM) fhwatsis.com/corsim page.htm.
14. (TRANSYT-7F) mctrans.ce.ufl.edu/featured/TRANSYT-7F.
15. (Trafficware-SimTraffic) trafficware.com/simtraffic.htm.