Coverage Density As a Dominant Property of Large-Scale Sensor Networks

Osher Yadgar[1] and Sarit Kraus[2*]

[1] SRI International, 333 Ravenswood Avenue Menlo Park, CA 94025-3493, USA, yadgar@ai.sri.com

[2] Bar Ilan University, Ramat Gan, Israel
sarit@cs.biu.ac.il

Abstract. Large-scale sensor networks are becoming more present in our life then ever. Such an environment could be a cellular network, an array of fire detection sensors, an array of solar receptors, and so on. As technology advances, opportunities arise to form large-scale cooperative systems in order to solve larger problems in an efficient way. As more large-scale systems are developed, there is a growing need to (i) measure the hardness of a given large-scale sensor network problem, (ii) compare a given system to other large-scale sensor networks in order to extract a suitable solution, (iii) predict the performance of the solution, and (iv) derive the value of each system property from the desired performance of the solution, the problem constraints, and the user's preferences.
The following research proposes a novel system term, the *coverage density*, to define the hardness of a large-scale sensor network. This term can be used to compare two instances of large-scale sensor networks in order to find the suitable solutions for a given problem. Given a *coverage density* of a system, one may predict the solution performance and use it jointly with the preference and the constraints to derive the value of the system's properties.

1 Introduction

On December 2, 2004, tsunami waves hitting the shores of Sri Lanka, India, Indonesia, and Thailand caused a loss of 300,000 lives and a tremendous tragedy for millions. Using current technology, sensors in the ocean could have sensed the creation and advancement of tsunami waves. A well-organized and well-managed network of such sensors using wireless communication could have produced an alarm that would have alerted control centers spread along the shores of the four countries. This alarm could have saved the lives of many of the victims. Studying the role of the different properties of such large-scale agent systems and tuning them according to the system constraints is in the core of this work. We will refer to large-scale agent systems capable of sensing objects as *large-scale sensor networks.*

* This work was supported in part by ISF grant no 8008 and by NSF under grant no. IIS0222914. Sarit Kraus is also affiliated with UMIACS.

Large-scale agent systems focus on the behavior of multiagent systems with many agents. As this is a relatively new research area, the number of agents needed to consider a multiagent system as a large-scale agent system has not yet been defined [28]. It became clear to us that researchers do not tend to refer to scale of large scale agent system by means other than the number of agents. Researchers may discuses system properties, but never include them as part of their large scale agent system definition. We claim that this system property is misleading. In this paper we prove that claim and suggest a better system property. We will introduce the *coverage density* and will show how this large-scale sensor network property can be used to predict the performance of a given instance of a large-scale agent system capable of sensing objects.

In the next section we survey related work. We will then introduce the *coverage density* term as a tool for classifying large-scale sensor network. Next we will demonstrate how to use the *coverage density* to predict the performance of a large-scale sensor network and to choose the values of the system properties. A simulator based on our previous work on large-scale sensor networks' architecture and challenges will be introduced. We will then report simulation results of a total of 50 years of CPU time examining hundreds of thousands of different agents and goals. We will conclude by showing how these results establish our claim that *coverage density* is a dominant property of large-scale sensor networks.

2 Related Work

Focusing on the number of agents forming large-scale agent system, the research community holds different opinions about how many agents may form a large-scale agent system. In some cases, thousands of agents are considered to be a large-scale agent system [7], [22], while in other cases hundreds compose such a system [3], [4], [23], [24] and there are even cases where only dozens of agents are considered to be a large-scale system [13].

We suggest that certain large-scale agent systems, i.e. large-scale sensor networks, should be measured relative to their context, such as the size of their problem space, and not simply by the number of participating agents.

Recent technology has made small low-cost devices available to build sensor networks [5], [8], [9], [10], [11], [27]. Wireless sensor networks benefit from technology advances in micro-electro-mechanical systems (MEMS) [25]. These advances have facilitated the development of low-cost simple wireless sensors. Combining the information gathered from thousands of such sensors is a very difficult problem [12]. However, solving this problem may lead to an efficient way of producing global information. As we have witnessed, such information could save lives. We will use the *coverage density* property to support this technological development. *Coverage density* may be used to (i) measure the hardness of a given large-scale sensor network problem, (ii) compare a given system to other large-scale sensor networks in order to extract a suitable solution, (iii) predict the performance of the solution, and (iv) derive the value of each system property from the desired performance of the solution, the problem constraints, and the user's preferences.

The term *sensor coverage* is traditionally used to denote the effectiveness of a sensor network [1], [15], [16], [17], [18], [14], [26]. Previous studies have shown that increasing the The *sensor coverage* increases the number of tracked objects. Gage [6] defines the coverage as a "spatial relationship that adapts to specific local conditions to optimize the performance of some function". Gage distinguishes between three basic types of coverage behavior: blanket coverage, where the objective is to achieve a static arrangement of sensors that maximizes the detection rate of targets appearing within the coverage area; barrier coverage, where the objective is to achieve a static arrangement of sensors that minimizes the probability of undetected targets passing through a barrier; and sweep coverage, where the objective is to move a group of elements across a coverage area to balance between maximizing the detection rate and minimizing the number of missed detections. Gage specifies that a sweep is equivalent to a moving barrier. The *coverage density* is different from the *sensor coverage*. The *coverage density* considers the many properties of a given system and is related to the problem space and time as well as to single-agent characteristics, while *sensor coverage* is related only to the geometrical alignment of the sensors.

3 The Coverage Density

Many properties influence the-scale and the hardness of a given problem. When designing a solution for a large-scale sensor network problem the different aspects of the system's properties should be considered. Properties such as the number of agents and the quality of the sensors, may be related to each other. For instance, given a limited budget, the system designer should consider whether to use many cheap sensors or a small number of expensive sensors. A classifying tool, the *coverage density*, is proposed to classify large-scale sensor networks and to predict the performance of different property configurations. This predicting tool can be used to design the system in order to meet imposed constraints such as budget limitations, battery supply, or technological constraints. The *coverage density* defines the time it takes to cover an area equal to the size of the controlled zone. The following definition formalizes this notation:

Definition 1. ***i.*** *let A be a set of n agents such that $A = \bigcup_{i \in N} a_i$ whereas $N = \{0, 1, ..., n-1\}$.*

ii. *let $w_{a_i}(t)$ be the area covered by the sensor of agent a_i at a given time t. The agent may detect objects in this area at time t.*

The measurement units of the area are square meters.

iii. *Let agent coverage, $\overline{w_{a_i}}$, be the average area covered by the sensor of agent a_i such that $\overline{w_{a_i}} = \frac{\int w_{a_i}(t) \cdot dt}{\int dt}$.*

The measurement units of the agent coverage are $\frac{\text{square meters}}{\text{second}}$.

iv. *Let total coverage,* $\overline{W_A}$*, be the average area covered by the sensors of all the agents such that* $\overline{W_A} = \sum_0^{n-1} \overline{w_{a_i}}$.

The measurement units of the total coverage are $\frac{\textit{square meters}}{\textit{second}}$.

v. *Let* Z *be the size of the controlled area.*

vi. *Coverage density* ρ *is the total coverage divided by the size of the controlled area such that* $\rho = \frac{\overline{W_A}}{Z}$.

The measurement units of the coverage density, ρ*, are* $\frac{1}{\textit{second}}$.

The *coverage density* denotes the amount of the controlled area that can be covered rather than the area that is actually covered. There may be an overlap of agent coverage such that, for example, a value of 100% *coverage density* does not reflect coverage of all the controlled area. A *coverage density* of 100% will result in full coverage only when the sensors have no overlapping coverage. We do not require such a constraint in sensor alignment.

4 Using The Coverage Density

To examine the role of the *coverage density* on large-scale sensor networks we used our hierarchical large-scale agent system architecture, the Distributed Dispatcher Manager (DDM) [19], [20], [21], and applied it to the Autonomous Negotiating Teams (ANTs) [2] problem.

4.1 The DDM Hierarchy

The DDM is designed for efficient coordinated resource management in large-scale agent systems; the model makes use of a hierarchical group formation to restrict the degree of communication between agents and to guide processes in order to very quickly combine partial information to form a global assessment. Each level narrows the uncertainty based on the data obtained from lower levels. DDM organizes the sensing agents in teams, each with a distinguished team leader agent. A team is assigned to a specific sector of interest. Each such agent can act autonomously within its assigned sector of interest while processing local data. Teams are themselves grouped into larger teams. Communication is restricted to flow only between an agent (or team) and its team leader agent. Each team leader is provided with an algorithm to integrate information obtained from its team members. Each individual information-collecting agent can extend its local information through the application of causal knowledge. The causal knowledge may be inaccurate and noisy, and therefore it only constrains the set of possible solutions that could be associated with a collection of data measurements.

In some cases naive distribution of thousands of agents will not be efficient, thus a load balancing mechanism may be required. Therefore, a load balancing mechanism is applied to the hierarchical architecture of the DDM. This mechanism dynamically balances the ratio between agents and goals throughout the controlled area. While applying the load balancing mechanism, the DDM strives to balance the ratio between agents and goals in each hierarchy level from the top down. Each level directs only its immediate subordinate level. The directed level, then, directs its own immediate subordinate level and so on. We will refer to DDM activating the load balancing algorithm as LB and to DDM not activating the load balancing algorithm as NLB.

4.2 The ANTs Challenge

We developed a simulator to study large-scale problems associated with the application of DDM. The ANTs challenge [2] was chosen as the test case of DDM, and simulated Doppler radars were selected as sensors. The key task of the ANTs challenge is to detect and track moving objects while using low-cost hardware. The simulation consists of an area of a fixed size in which Doppler sensors attempt to extract the object state functions of moving targets. Doppler sensors are attached to mobile agents named samplers. The ANTS program uses Doppler sensors that may activate its beam in three different directions. According to the ANTS specific Doppler radar, only one direction may be activated at a time. The orientations of the beams are 0, 120 and 240 degrees. Each of these sensors may move and spin around its center. Given a measurement of a Doppler radar the target is located based on the following equation:

$$R_i^2 = \frac{k \cdot e^{\frac{-(\theta_i - \beta)^2}{\sigma}}}{\eta_i} \tag{1}$$

Where, for each sensed target, i, R_i is the distance between the sensor and i; θ_i is the angle between the sensor and i; η_i is the measured amplitude of i; β is the sensor beam angle; and k and σ are characteristics of the sensors and influence the shape of the sensor detecting area.

A sampler agent may sense targets only when it is not moving. While sensing, a sampler agent may detect targets located within a short distance from it. We refer to this distance as the *range of interaction*. The number of sampler agents, the time spent on detection versus the time spent on movement, and the range of interaction characterize the system. As we will see, these characteristics determine the quality of the solution. Balancing these characteristics results in a desired system quality under given limitations, such as budget.

5 Experiments

One hundred and fifty different personal computers running Windows XP, Windows 2000, and Linux operating systems were used to simulate hundreds of scenarios for four consecutive months. Each scenario simulated 7 days of target tracking. A total of 50 years of CPU time were logged, examining hundreds of thousands of different agents and goals.

While using the basic settings (Table 1) we simulate a 400,000,000 square meter area. In this area, agents track moving targets. At a given time 1,000

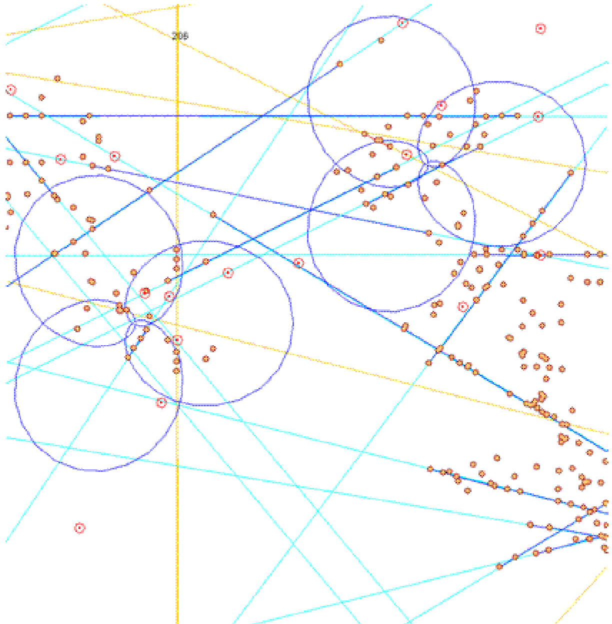

Fig. 1. Simulation of two sensors

targets are moving in the area. In total, during 7 days, 13,635 targets enter and exit the controlled area at any given time. Each target has an initial random location along the border and an initial random velocity as high as 50 kilometers per hour in a direction that leads inward. Targets leave the area when reaching the boundaries of the area. Each target that leaves the area causes a new target to appear at a random location along the border and with a random velocity in a direction that leads inward. Therefore, each target may remain in the area for a random time period.

There are 5,000 Doppler agents and each of them contains a Doppler with a beam range of 50 meters. Since the Doppler beam is essentially a circle while the range of interaction is its diameter (Figure 1), the area covered by the beam is 1,963 square meters. Every Doppler senses targets for 10 seconds and then moves for 5 seconds. That is, on the average, the *agent coverage* , $\overline{w_{a_i}}$, is 1,309 $(\frac{m^2}{s})$. Using 5,000 similar Dopplers leads to a *total coverage*, $\overline{W_A}$, of 6,544,985 $(\frac{m^2}{s})$. Dividing the *total coverage* by the size of the controlled zone results in a *coverage density* of 1.64%. In each experiment, we vary one of the parameters of the environment, keeping the other values of the environment parameters as in the basic settings.

As stated earlier, the *coverage density* ρ characterizes the size of a given large-scale sensor network problem. As ρ decreases, the size of the problem increases. In the following sections we present a study on the properties that affect ρ. We will show how different sets of properties, having similar *coverage density*

Table 1. Experiment basic settings

Z (m^2)	400,000,000
Agent properties	
Range of interaction (m)	50
Sensing time (sec)	10
Moving Time (sec)	5
$w_{a_i}(t)$ when sensing (m^2)	1,963
$w_{a_i}(t)$ when moving (m^2)	0
$\overline{w_{a_i}}$ ($\frac{m^2}{sec}$)	1,309
Number of agents	5,000
$\overline{W_A}$ ($\frac{m^2}{sec}$)	6,544,985
ρ ($\frac{\mathbf{1}}{\mathbf{sec}}$)	**1.64%**

values, achieve similar results. We will establish a strong correlation between the *coverage density* and system performance as opposed to the other properties, such as the number of agents.

We will begin studying the *coverage density* by changing the number of sensing agents. As the number of sensing agents increases, the *coverage density* grows (Table 2 - left). The number of sensing agents not only influences the *coverage density* but also holds an important role in distributing the solution and reducing the computation load. We will show that up to a certain point, decreasing the *coverage density* by reducing the number of agents slightly affects the amount of tracked targets. In our load balancing algorithm case, having only 3,000 agents, which is a *coverage density* of 0.98%, yields results almost equivalent to the results of situations with a much larger number of agents. Having only 3,000 agents moderately reduces the tracked target percentage in comparison to not using the load balancing algorithm. Using fewer agents will reduce the tracked target percentage more dramatically. After focusing on the impact that changing the quantity of agents has on the system, we proceeded to study the impact of changing the quality of the sensor agents.

Table 2. How changing one property while keeping the other as in the basic settings influences the *coverage density*

Agent population		Range of interaction		Moving vs. sensing time	
value	ρ	value	ρ	value	ρ
1000	0.33%	13	0.11%	2:1	0.82%
2000	0.65%	25	0.41%	* **1:2**	**1.64%**
3000	0.98%	* **50**	**1.64%**		
4000	1.31%	100	6.54%		
* **5000**	**1.64%**	200	26.18%		
7000	2.30%				
8000	2.63%				
9000	2.95%				

* **belongs to the basic settings**

The range of interaction in the basic settings was 50 meters. In the following section we compare the behavior of DDM while using sensors with different ranges of interaction. This range reflects the complexity and the cost of the sensors. As the range of interaction increases, the sensor is likely to be more complex and expensive. Therefore, there may be an interest in using sensors with smaller ranges of interaction. However, decreasing the range of interaction increases the *coverage density* and therefore increases the scale of the problem. We will show how decreasing the *coverage density* by reducing the range of interaction influences DDM performance with and without the load balance algorithm (Table 2 - middle).

The next property affecting the *coverage density* is the ratio between the sensing time and the moving time (Table 2 - right). As in the range of interaction, the ratio affects the cost of using DDM. Activating the sensor for longer periods of time is likely to cost more if the energy of producing the electromagnetic beam is expensive. On the other hand, a more mobile sensor may cost more if the fuel needed to drive a sensor around the controlled zone is expensive. Activating the sensor for longer periods of time at the expense of movement time decreases the *coverage density* and therefore reduces the scale of the problem. We will show that this period of time influences the performance of DDM with and without the load balancing mechanism. We will also show how to compensate for using simple and less expensive sensors by increasing the period of the sensing time.

6 Results

In the following sections we will show that system performance is strongly correlated to the *coverage density* rather than to the number of agents, the range of interaction, and the sensing/moving time. In Figures 2 ,3 ,4 the *coverage density* value is included below the X axis in parentheses. Results of DDM applying load balancing are in black, while DDM without load balancing is in gray. The results of the basic setting are denoted in bold symbols.

6.1 Agent Population

We investigated the influence of the *coverage density* through changing the number of sensor agents. During this investigation we ran different scenarios. Each scenario had the same properties (see Table 1) with a different number of sensor agents. In the first scenario, there were 1,000 agents; in the second, 2,000 agents; in the third, 3,000 agents; in the fourth, 4,000 agents; in the fifth, 5,000 agents; in the sixth, 7,000 agents; in the seventh, 8,000 agents; and in the eighth, 9,000 agents. The *coverage densities* of the scenarios were 0.33%, 0.65%, 0.98%, 1.31%, 1.64%, 2.30%, 2.63%, and 2.95%, respectively, whereas the basic setting trial had a *coverage density* of 1.64%.

Following Figure 2, one can see an improvement in performance as the number of agents increases. Note that an increment of the number of agents reflects a corresponding increment of the *coverage density* (Table 2). The improvement

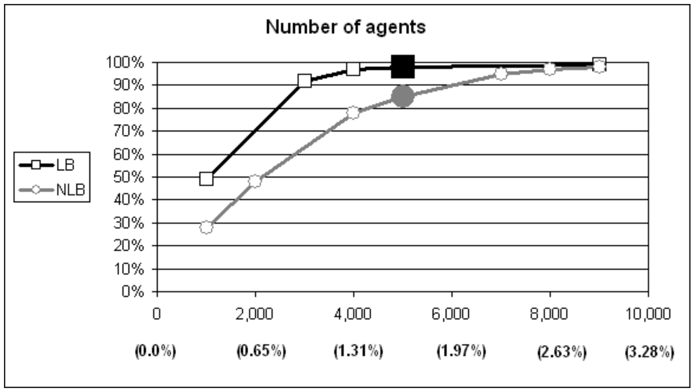

Fig. 2. Tracking percentage as a function of the number of agents

is achieved regardless of whether the a load balancing mechanism is used. The results show that the system can efficiently utilize additional resources. However, the improvement is significant only up to a certain number of agents. For NLB it is 7,000 agents and for LB it is 4,000. In the next paragraphs we will show that the major influence on the system performance is the *coverage density* and not just the number of agents.

6.2 Range of Interaction

Another property influencing the *coverage density* is the sensor maximum range of interaction. In this part of the study we varied the range of interaction and compared the performance with and without load balance. In different settings, the maximum detection range of each sensor was 13, 25, 50,100, and 200 meters. This translates to coverage densities of 0.11%, 0.41%, 1.64%, 6.54%, and 26.18%, respectively.

In Figure 3 one can see that the system performs better as the range of interaction increases. That may be explained by the fact that each sensor may detect more targets as its range increases. Once again, the improvement is presented for both NLB and LB cases and is significant up to a certain point. We can see that keeping the number of agents constant does not imply that the performance of the system will stay constant. However, as in the case of the number of agents, increasing the range of interaction reflects a corresponding increment of the *coverage density* (Table 2). We can see that in both cases increasing the *coverage density* leads to better performance.

6.3 Sensing and Moving Time

In the basic settings each agent repeats the following activities: (i) it senses targets for 5 seconds and then (ii) it moves for 10 seconds. The ratio between the moving time and the sensing time is therefore 1:2 in the basic setting case.

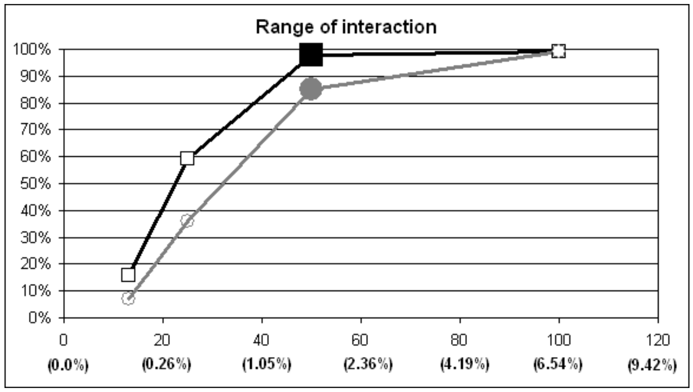

Fig. 3. Tracking percentage as a function of the range of interaction

In this case the *coverage density* was 1.64%. To check the impact of the *coverage density* on the ratio between the sensing and the moving times we compared the basic settings to settings with different sensing and moving times. We increased the sensing period to 20 seconds while the moving period remained at 10 seconds. The ratio between the sensing time and the moving time in this case was 2:1 and the *coverage density* was 0.82%. The fact that the *coverage density* in this setting was lower than for the basic settings suggests that the problem of accurately identifying the targets' trajectories is harder.

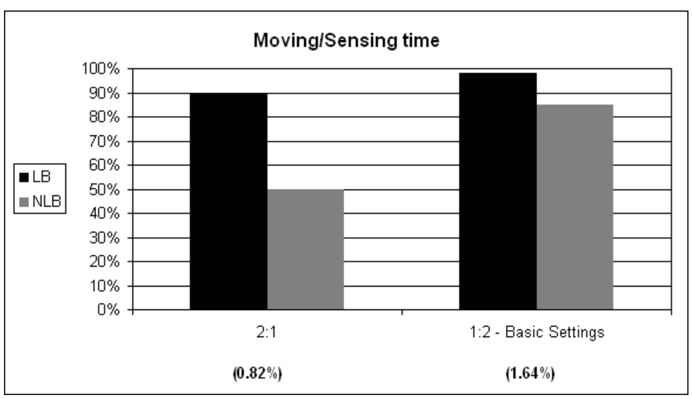

Fig. 4. Tracking percentage as a function of the sensing-moving ratio

Looking at Figure 4 we can see that once again keeping the number of agents constant does not ensure achieving the same performance. Moreover, it supports our findings that decreasing the *coverage density* decreases performance. In this case we decreased the *coverage density* from 1.64% to 0.82%, and the performance dropped from 98% to 90% while we applied a load balancing and from 85% to 50% while we did not.

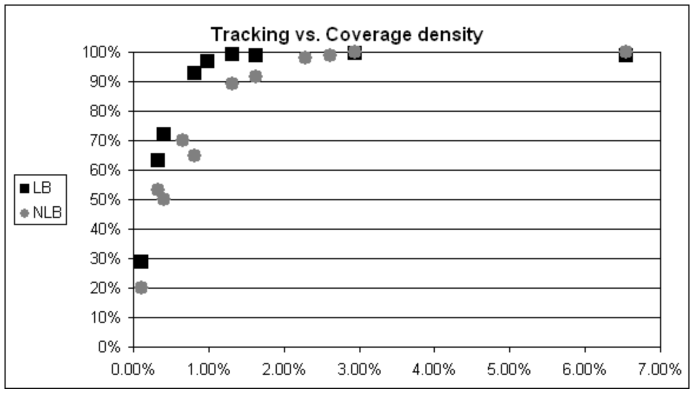

Fig. 5. Integrated results: performance vs. the *coverage density*

6.4 Integration

To establish the importance of the *coverage density* definition we compared the tracking percentage of different settings. Figure 5 presents the results reported in the previous paragraphs as a function of the *coverage density*. The results are the aggregation of the results presented above. Looking at Figure 5, one cannot distinguish between the different scenario sets. The results of all the sets are situated along the curve of the LB and the NLB graphs. This proves that the important characteristic of large-scale sensor networks is the *coverage density* and not a single component of it. Moreover, Figure 6 demonstrates that the number of agents does not necessarily predicts system performance. When we used 5,000 agents with different ranges of interaction or sensing/moving times, the system performed differently. The same results presented in Figure 6 were measured for the sensing/moving time and the range of interaction properties.

To compare the correlation between system performance and its properties, we calculated the correlation coefficient of each property. We used only the measurements that described a contribution to system performance. For example, we considered all the results up to 5,000 agents for the *Agent population* property with load balancing (see Figure 2). We did that because there is no sense in looking for correlation after the system reaches its full utilization. The correlation coefficient for the *Agent population* property with load balancing was 0.2 and without load balancing was 0.32. For the *Range of interaction* property with load balancing the coefficient was 0.79, while without the load balancing it was 0.68. The *Sensing/moving time* property had the worst correlation of 0.08 with load balancing and -0.16 without load balancing. The best correlation was

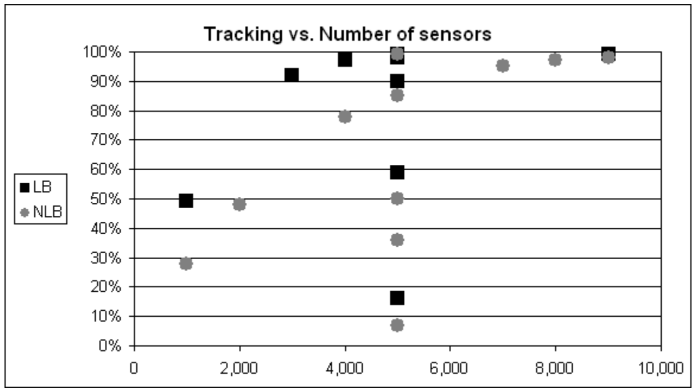

Fig. 6. Integrated results: number of agents

achieved by the *coverage density* and had the value of 0.86 with load balancing and 0.89 without load balancing. These results prove that the *coverage density* has a strong correlation to system performance.

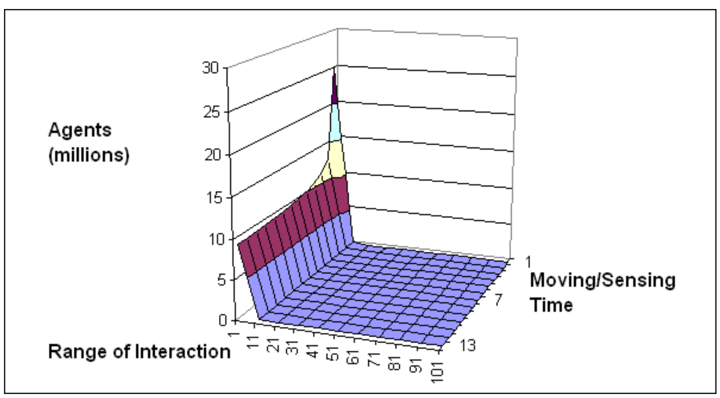

Fig. 7. Different settings that have the coverage density of the basic settings (1.64%)

To further depict the uses of the *coverage density*, Figure 7 presents the different alternatives for the basic settings. The figure illustrates an isogram surface of a continuous variation of the three property dimensions. Recalling that the *coverage density* is correlated to the multiplication of these three properties, each point on the surface represents a certain number of agents, an interaction range, and a moving-sensing ratio that result in the same *coverage density* of 1.64%, which is the same *coverage density* of the basic settings.

7 Conclusions

Coverage density defines the time needed to cover an area equal to the size of the controlled zone. We have shown that there is a strong correlation between the *coverage density* of a system and its behavior. In comparing large-scale sensor networks having different *coverage densities*, we have proven that system properties such as the number of agents, the range of detection of each agent, and the agent's activation time have the same influence on the number of detected objects. For instance, by analyzing only the number of objects a large-scale sensor network successfully detects, one may not know whether there is a large number of cheap sensors or a small number of expensive ones. Given this fact, we introduced a way to achieve the same system results with different preferences. As a result, a system designer may find it easier to achieve a certain level of system performance under given specific constraints, such as budget limits.

References

1. A. Howard, M. J Matarić and G. S. Sukhatme, 'Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem', *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, pages 299-308, Fukuoka, Japan, 2002.
2. Autonomous Negotiating Teams (ANTs) website, *Darpa*, http://www.rl.af.mil/tech/programs/ants/
3. C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann and F. Silva, 'Directed Diffusion for Wireless Sensor Networking', *IEEE/ACM Transactions on Networking*, **11(1)**: 2-16, February 2003.
4. C. Ortiz, K. Konolige, R. Vincent, B. Morisset, A. Agno, Mi. Eriksen, D. Fox, B. Limketkai, J. Ko, B. Steward and D. Schulz, 'Centibots: Very Large Scale Distributed Robotic Teams', *AAAI*, 1022-1023, 2004.
5. D. Ganesan, A. Cerpa, Y. Yu, W. Ye, J. Zhao and D. Estrin, 'Networking Issues in Sensor Networks', *Journal of Parallel and Distributed Computing (JPDC)*, Special Issues on Frontiers in Distributed Sensor Networks, **64(7)**, 799–814, 2004.
6. D.W. Gage, 'Command Control for Many-Robot Systems', *Proceedings of AUVS92, 19th Annual Technical Symposium and Exhibition of the Association for Unmanned Vehicle Systems*, Huntsville AL, pages 22-24, June 1992.
7. E. Ogston, B. Overeinder, M. van Steen and F. Brazier, 'A Method for Decentralized Clustering in Large Multi-agent Systems', *AAMAS*, pages 789-796, 2003.
8. G. Asada, M. Dong, T. S. Lin, F. Newberg, G. Pottie, H. O. Marcy and W. J. Kaiser, 'Wireless Integrated Network Sensors: Low Power Systems on a Chip', *IEEE ESSCIRC '98*, pages 9-16, September 1998
9. G.J. Pottie and W.J. Kaiser, 'Wireless Integrated Network Sensors', *Communications of the ACM*, **43(5)**: 551-558, 2000.
10. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, 'Wireless Sensor Networks: A Survey', *Computer Networks* **38(4)**: 393-422, 2002.
11. J. Liu, L. F. Perrone, D. M. Nicol, M. Liljenstam, C. Elliott and D. Pearson, 'Simulation Modeling of Large-scale Ad-Hoc Sensor Networks', *Proceedings of Euro-SIW01, European Simulation Interoperability Workshop*, 2001.

12. K. Bult, A. Burstein, D. Chang, M. Dong, M. Fielding, E. Kruglick, J. Ho, F. Lin, T. H. Lin, W. J. Kaiser, H. Marcy, R. Mukai, P. Nelson, F. Newberg, K. S. J. Pister, G. Pottie, H. Sanchez, O. M. Stafsudd, K. B. Tan, C. M. Ward, S. Xue and J. Yao, 'Low Power Systems for Wireless Microsensors', *Proceedings of International Symposium on Low Power Electronics and Design*, pages 17-21, 1996.
13. K. H. Low, W. K. Leow and M. H. Ang Jr., 'Task Allocation via Self-Organizing Swarm Coalitions in Distributed Mobile Sensor Network',*AAAI*, pages 28-33, 2004.
14. K. Kar and S. Banerjee, 'Node Placement for Connected Coverage in Sensor Networks', *Proceedings of WiOpt 2003: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, March 2003.
15. M. A. Batalin and G. S. Sukhatme, 'Multi-robot Dynamic Coverage of a Planar Bounded Environment', *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.
16. M. A. Batalin and G. S. Sukhatme, 'Spreading Out: A Local Approach to Multi-Robot Coverage', *Proceedings of 6th International Symposium on Distributed Autonomous Robotic Systems*, 373-382, 2002.
17. M. A. Batalina and G. S. Sukhatmea, 'Sensor Coverage using Mobile Robots and Stationary Nodes', *Proceedings of the SPIE*, volume 4868: 269-276, Boston, MA, August 2002.
18. M. Cardei and J. Wu, 'Coverage in Wireless Sensor Networks', *Handbook of Sensor Net*, M. Ilyas and I. Mahgoub (eds), CRC Press, 2004.
19. O. Yadgar, 'Efficient Algorithms in Large-scale Agent Systems', PhD Thesis, Bar-Ilan University, 2005.
20. O. Yadgar, S. Kraus and C. Ortiz, 'Scaling Up Distributed Sensor Networks: Cooperative Large-scale Mobile-Agent Organizations', in *Distributed Sensor Networks: A Multiagent Perspective*, pp. 185-218, Kluwer publishing, 2003.
21. O. Yadgar, S. Kraus and C. Ortiz, 'Hierarchical Information Combination Process for Large-scale Task and Team Environments', in *Communications in Multiagent Systems*, Springer-Verlag, 2003.
22. P. J. Turner and N. R. Jennings, 'Improving the Scalability of Multi-agent Systems', *Proceedings of the First International Workshop on Infrastructure for Scalable Multiagent Systems*, Springer-Verlag, pages 246-262, Barcelona, Spain, June 2000.
23. P. Scerri, R. Vincent and R. Mailler, 'Comparing Three Approaches to Large Scale Coordination', *Proceedings of the First Workshop on the Challenges in the Coordination of Large Scale Multi-agent Systems*, July 2004.
24. P. Scerri, Y. Xu, E. Liao, J. Lai and K. Sycara-Cyranski, 'Scaling Teamwork to Very Large Teams', *AAMAS*, pages 888-895, 2004.
25. R. Niu, P. Varshney, M.H. Moore and D. Klamer, 'Decision Fusion in a Wireless Sensor Network with a Large Number of Sensors', *Proceedings of the Seventh International Conference on Information Fusion*, Stockholm, Sweden, June 2004.
26. S. Shakkottai, R. Srikant and N. Shroff, 'Unreliable Sensor Grids: Coverage, Connectivity and Diameter', *Proceedings of the IEEE INFOCOM*, pages 1073-1083, 2003.
27. T. H. Lin, H. Sanchez, R. Rofougaran and W. J. Kaiser, 'CMOS Front End Components for Micropower RF Wireless Systems', *Proceedings of the 1998 International Symposium on Low Power Electronics and Design*, pages 11-15, 1998.
28. V. Lesser, C. Ortiz and M. Tambe (ed), 'Distributed Sensor Networks: A Multiagent Perspective', Kluwer Publishing, 2003.