Utility-Based Joint Sensor Selection and Congestion Control for Task-Oriented WSNs

Hosam Rowaihy*, Matthew P. Johnson§, Sharanya Eswaran* Diego Pizzocaro‡,
Amotz Bar-Noy§, Thomas La Porta*, Archan Misra† and Alun Preece‡
*Department of Computer Science and Engineering, Pennsylvania State University, USA
§Department of Computer Science, The Graduate Center, City University of New York, USA
‡School of Computer Science, Cardiff University, UK
†Telecordia Technologies, Inc., USA

***Abstract*—Task-centric wireless sensor network environments are often characterized by the simultaneous operation of multiple tasks. Individual tasks compete for constrained resources and thus need resource mediation algorithms at two levels. First, different sensors must be allocated to different tasks based on the combination of sensor attributes and task requirements. Subsequently, sensor data rates on various data routes must be dynamically adapted to share the available wireless bandwidth, especially when links experience traffic congestion. In this paper we investigate heuristics for incrementally modifying the sensor-task matching process to incorporate changes in the transport capacity constraints or feasible task utility values.**

I. Introduction

Sensor networks are typically required to do multiple simultaneous task. These tasks do not only compete for the sensing resources available in the field but also for the wireless links in the network. Given that both resources are constrained, mediation algorithms become necessary. Prior works have used utility-based approaches to address the sensor task matching problem and link rate assignment problems in task-oriented wireless sensor networks independently. Algorithms for both optimal or near-optimal assignment of sensors to competing tasks of multiple priorities and, separately, for applying distributed network utility maximization (NUM) techniques to mitigate network congestion were developed.

In this work, we address the problem of joint treatment of these two separate problems. We define an iterative approach where the sensor assignment algorithms operate on a slower time scale with awareness of the currently achieved task satisfaction levels, which are themselves computed by a faster time scale adaptation of sensor rates based on transport capacity constraints. Our work demonstrates how convergence time and overall utility may be improved by sharing information across the matching and rate adaptation processes.

We propose two algorithms for assigning sensors to tasks. Both algorithms try to select the sensors that provide the best sensing utility to the tasks. They also both run in rounds to reach the best possible assignment. They differ, however, in the way and the order in which they select sensors if, due to contention on edges, the best sensors cannot provide acceptable rates to the tasks' sinks.

The first algorithm, called *Greedy-Divide*, only uses information about the interference along the routes from selected sources to the sinks for the different tasks. NUM is used after acceptable interference levels are reached at which time it decides on the optimal rates for the selected topology. The second algorithm, called *NUM-based Iterative*, runs NUM in rounds and using the rate results to decide on which task is affected the most by the contention on links and hence should reselect sensors. We find through simulation that both algorithms improve the network utility by 5% to 14% compared to assigning sensors based only on the utility a sensor provides to a task without considering the rates.

We also propose three schemes for base station selection. This is motivated by our observation that contention on links causes the most loss in terms of utility. This loss is higher when each task has a single base station to which all data is delivered. We introduce the use of multiple base stations which are connected to the sinks to alleviate this problem.

II. Related Work

Sensor Selection. There has been work on defining frameworks for sensor-task assignment. For example, [1] defines a framework for the assignment problem in which the goal to maximize the utility while staying under a predefined budget. The general problem of sensor selection to achieve an objective has also received sizable attention lately. For example, in [2], [3] the authors solve the coverage problem, which is a related problem, using the least number of sensors to conserve energy. The problem we consider here considers multiple tasks that contend for the same set of sensors which calls for resolution mechanisms. The Semi-Matching with Demands (SMD) problem for sensor-task assignment was recently introduced in [4] and was later extended in [5] and [6].

Congestion Control. There is a wealth of literature on optimization-based techniques for rate control in communication networks, an approach first introduced by Kelly *et al.* [7], [8] for the case of *wired links*. In this model, each source node s is associated with a concave, non-decreasing utility function, which depends on the source's transmission rate. The network is modeled as a set of links and each flow is a collection of links. This classical network utility maximization (NUM) framework was recently extended in [9] and [10] to a more general WSN environment, where the topology is arbitrary and individual sensors have subscribing tasks. In this extended model, each task derives its utility from a composite set of sensors assigned to it. Intermediate nodes are used to forward sensor data to tasks' data sinks.

III. Network Model

We assume a set of static sensors pre-deployed in a field. The deployed sensors are directional in nature and hence each of them can be assigned to a single task (i.e. directed to one location). They also have limited sensing range R_s; only sensors that are within that range from a task can be assigned to the task. We assume that sensors use the same communication equipment and hence link capacities are equal. Sensors are aware of their location.

Multiple sensing tasks that require information, from one or more deployed sensors, exist in the system. Each task is defined by a specific geographic location and has a sink. Data gathered by sensors in the field are transmitted over a multihop route, through sensors acting as relays, to a base station from which it is transferred through a wired network to the sink for that task. The routes from sensors to the base stations are pre-determined.

IV. Problem Definition

In this section we define the problem assuming that each task has a sink that is attached to a unique base station. Later in the paper (see Section VI) we relax this assumption and decouple sinks from base stations. This allows for an extra level of freedom in making assignment decisions.

Let $\{S_i : i = 1 \dots n\}$ be a set of sensors and $\{T_j : j = 1, \dots, m\}$ be a set of tasks. The sensors and tasks appear as nodes in some larger graph, which also includes relay nodes. The simplest relevant matching problem is weighted bipartite matching (also known as the assignment problem), though we relax the matching requirement to allow many-one sensor assignment. Assuming flows f_{ij} on edges g_{ij}, the problem is to choose a node-disjoint set of edges of maximum total flow.

Conversely, we have the NUM problem. In a simple version of this problem, a set of $S_i \rightarrow T_j$ paths or routes r_{ij} are given. Also given are capacity constraints limiting the feasible flow on each edge in the network. Each edge g has a capacity c_g which, due to interference, depends on the number of flows sharing the same edge. The maximum potential flow of a pair (i, j) is the minimum edge capacity along the route r_{ij}, i.e. $C_{ij} = \min_{g \in r_{ij}} c_g$. The task is to assign a feasible flow to each route, maximizing total flow.

The two problems can be combined as follows. Given the network, including specified sensors and task sinks, the two problems are, when viewed in sequence, first to choose an assignment of sensors to tasks and then to choose the routes' flow values. Moreover, we can add utility values e_{ij} to the routes. The utility of the route represents the relevance of the sensor's information to the task, in contradistinction to the amount of information allowed to flow on this route f_{ij}. This core problem, which we refer to here as NUM/Matching, can be formulated as a mixed integer program (MIP):

$$\begin{array}{ll} \textit{max:} & \sum_{i,j} e_{ij} f_{ij} \\ \textit{s.t.:} & \sum_{r_{ij}:\ g \in r_{ij}} f_{ij} \leq c_g \text{ , for each edge } g, \\ & f_{ij} \leq C_{ij} x_{ij}, \text{ for all } (i,j) \\ & \sum_j x_{ij} \leq 1, \text{ for all } S_i \\ & x_{ij} \in \{0,1\},\ f_{ij} \geq 0 \end{array}$$

In this formulation, there are two sets of variables, x_{ij} indicating whether route r_{ij} is selected, and f_{ij} indicating the flow along this route (if selected), which is weighted in the objective function by the profit value. We assume each route's maximum potential flow is encoded in the network itself. The first two sets of constraint enforce network capacities. The remaining constraints prevent any sensor from being assigned to more than one task. Note that the second constraint set ensures that flow $f_{ij} = 0$ if the decision variable $x_{ij} = 0$.

If we drop the capacity constraints, then we are left with the assignment problem. If we drop the unique-assignment constraints, then we are left with a constrained version of max-flow. Both of these problems (each solvable optimally in polynomial time) can therefore be viewed as relaxations of the problem. The *minimum* of their solutions provides an upper bound on the optimal solution of the combined problem. Unfortunately, solving the problem itself optimally is computationally hard.

Theorem 1: The NUM/Matching problem is NP-hard, even in the special case of unit edge capacities and unit profits.

Proof: We reduce from 3-SAT [11]. In that problem, we are given a boolean formula of the form $A_1 \wedge A_2 \wedge ... \wedge A_n$, where each clause A_i is of the form $\ell_1 \vee \ell_2 \vee \ell_3$, where each ℓ_j is a variable or a negated variable. The task is to determine whether the formula is satisfiable.

Given a 3-SAT instance, we produce a NUM/Matching instance with n sensors and $3n$ tasks, each of profit value one. For each clause A_i, we create a sensor named S_i and three tasks, labeled $(i, \ell_{i,1}), (i, \ell_{i,2}), (i, \ell_{i,3})$, where $\ell_{i,j}$ indicates the jth literal appearing in clause A_i. These three tasks each connect to S_i on routes of capacity 1. Note that each sensor has only one non-zero-value route.

We now explain how to construct these routes. We require that there be a *conflict* between any two routes for tasks (i_1, ℓ_{i_1,j_1}) and (i_2, ℓ_{i_2,j_2}) if ℓ_{i_1,j_1} is the negation of ℓ_{i_2,j_2} or vice versa. This means that these two routes share an edge, which implies that the sum of their flows cannot be greater than one. Note that the total number of conflicts (and hence the number of additional edges inserted) is bounded by $O(n^3)$.

We claim that the constructed instance has an optimal solution of value n iff the 3-SAT instance is satisfiable. If the optimal value is n, then by the matching constraint and the unit capacities and profits, n of the tasks each receive value one. By construction, the only way for a task to receive value one is for its unique route to have flow exactly one. Hence any route conflicting with this flow must have value zero, which implies the formula is satisfied. Conversely, if the formula is satisfied, then in each clause there is a least one literal which can be made true, and so at least one route for the corresponding sensor that can be given flow one. ■

Given the hardness of the problem we turn to heuristics

Algorithm 1 Greedy-Divide

for each task T_j
  while($|S(T_j)| < N$)
    $S(T_j) \leftarrow S(T_j) + \{max(S_i : e'_{ij})\}$
    $e'_{ij} = e_{ij}/(1$ + *max. interferers along route*)
$r \leftarrow 0$
while($r < R$ and $decrease \geq \alpha$)
  Find task T_j with highest number of interferers
  Find source $S_i \rightarrow T_j$ with maximum interferers
  Replace S_i with $max(S_k : e'_{kj})$
  $r \leftarrow r + 1$
Run NUM

Algorithm 2 NUM-based Iterative

for each task T_j
  while($|S(T_j)| < N$)
    $S(T_j) \leftarrow S(T_j) + \{max(S_i : e_{ij})\}$
$r \leftarrow 0$
while($r < R$ and $increase \geq \beta$)
  Run NUM
  Find task T_j with minimum rate utility U_{r_j}
  Find source $S_i \rightarrow T_j$ with maximum interferers
  $e'_{ij} = e_{ij}/(1$ + *max. interferers along route*)
  Replace S_i with $max(S_k : e'_{kj})$
  $r \leftarrow r + 1$

to provide solutions. In the rest of the paper, we consider a generalization of this problem to more complex utility models, still with the objective of maximizing the sum total of the utility received by the tasks in the network.

V. Algorithms

In this section we provide the details about the two proposed algorithms. Although the algorithms differ in the way and the order in which they decide which sensors to choose, they are similar in some aspects of the protocol they follow.

A. Overview

For each task, a *leader* is chosen, i.e. a sensor close to the task's location. Each task leader is informed about the location of its task by a command center. Task leaders then run a local algorithm to match nearby sensors to the requirements of the task. We allow a task to assign up to N sensors. The set of assigned sensors to task T_j is denoted by $S(T_j)$.

Since the utility a sensor can provide to a task is limited by a finite sensing range (R_s), only nearby sensors are considered. The leader advertises its task information to the nearby sensors (e.g. sensors within a certain number hops).

To decide on the actual rates at which each source sends data we run NUM. This occurs after the final assignment in the Greedy-Divide algorithm and in each iteration of the NUM-based iterative algorithm. Running NUM also determines the rate utility of each task T_j denoted by U_{r_j}.

B. Greedy-Divide

In this algorithm (shown as Algorithm 1), we make a first approximation of the competition on a link before making an assignment. Before an assignment is made, sensor S_i uses (1) its actual utility to task T_j, e_{ij}, and (2) the number of interfering flows along the route to the base station, to determine its *effective utility*, e'_{ij}. The effective utility is equal to the actual utility divided by the maximum number of interferers, i.e. the bottleneck, along the route to the base station. In this way, sensors attempt to account for competition on links before proposing to a task. A task selects sensors greedily based on the effective utility.

After the initial assignment, the algorithm runs for R rounds. In each round, the task with the highest number of interferers (calculated as the sum of the interferers of all sources) releases the sensor that has the most interfering flows along its route. The task leader reselects a sensor based on the updated effective utilities of the nearby sensors. The information about which task is facing the most interference can be learned by overhearing or by limited flooding of interference information. To calculate its effective utility, each sensor that can serve the task probes the network to determine the number on interferers along the route.

To allow the algorithm to progress beyond a task that has no better options, we limit the unsuccessful retries that a task can have to t tries. An unsuccessful try is a reselection that leads to a decrease or no change in the task's number of interfering flows. The algorithm can terminate earlier than the R rounds if no significant improvement is seen. The termination point is determined using an application parameter α. That is if there is less than α percent decrease in the total number of interferers of all tasks then the algorithm terminates.

C. NUM-based Iterative

Initially, this algorithm (shown as Algorithm 2) simply assigns sensors to tasks in order of the utility they provide. We do not consider possible competition on the links when making the initial assignments, so it is likely the utility achieved after running NUM will be substantially lower than expected. After the initial assignment, we run for R rounds.

In each round, we first run NUM to set the rates of the links given the current assignment. Then, the task that achieves the lowest *rate* utility based on the results of NUM releases the sensors that is facing the most interference. After that, the task reselects a sensors based on the updated effective utilities of the nearby sensors, i.e. taking the number of interferers into account as in the previous algorithm.

Again, to allow the algorithm to progress beyond a task that has no better options, we limit the unsuccessful retries that a task can have to t tries. Here, an unsuccessful try is a reselection that leads to decrease or no change in the task's rate utility as determined by NUM. Also, the algorithm can terminate earlier than the R rounds if less than β percent increase in the overall utility of the network is achieved.

The main difference between this algorithm and Greedy-Divide is that the decision on which task should reselect is based here on a centralized process that requires high communication overhead to reach the optimal rates. Greedy-Divide, on the other hand, can gather the needed information using overhearing or more efficient limited flooding.

VI. Base Station Selection

In this section we propose three methods for a source to choose a base station. Given that the routing table is static in the network, the route from a sensor to the base station must always be the same. Because of congestion, however, we introduce the concept of having multiple base stations that are connected to all the task sinks.

There are three ways for a source to select a base station:

1) **Single base station:** Each task has a unique base station to which data from all sources are sent.
2) **Static base station selection:** Multiple base stations are deployed. A sensor always chooses the base station that has the shortest route.
3) **Dynamic base station selection:** Multiple base stations are deployed. A source chooses the base stations dynamically based on which one has the lowest number of interferers along the route. This decision can change dynamically as the state of the network changes.

Note that in the second and third case, we assume that all the base stations are connected to task sinks. Hence, a source can be assigned to any base station and the data is delivered to the appropriate sink. The last link from the base station(s) to the sink is assumed to be wired and hence we do not consider interference issues at this last hop.

VII. Performance Evaluation

In this section we show the simulation results comparing the performance of the the *Greedy-Divide* (**G-D**) and the *NUM-based Iterative* (**NBI**) algorithms. NUM is run after each round to determine the overall performance of the network. The application we use is event detection in which the goal is to assign sensors that can provide the highest combined detection probability given that utility may change due to the sensor sending data at a lower rate than expected.

In the experiments, 500 sensors are uniformly deployed in a field that is $250m \times 250m$ in area. There are also 16 base stations that are deployed based on a uniformly random distribution. Tasks are created in uniformly distributed locations in the field. In the first experiment, there are 16 tasks each assigned a single sensor. In the second experiment, there are 4 tasks each assigned 4 sensors. In both cases the number of assigned sensors is equal to 16.

A. Assumptions

The utility of a sensor S_i to task T_j or e_{ij} represents the probability of sensor S_i detecting an event at T_j's location. The detection probability model we use is as follows: we divide the circle of radius equal to the sensing range R_s (centered at the task's location) into rings. Sensors within the same ring are assumed to be able to detect an event at the center with the same probability. In our experiments, we use R_s = 40m and divide the circle into 4 rings. Sensors in the inner circle provide 90% probability of detection, and for those within the second ring it becomes 80%, and so on. Sensors within the outermost ring provide 60% detection probability. The cumulative detection probability that a task achieves from the assigned sensors ($\{S_i \rightarrow T_j\}$) is what we call the *assignment utility* (U_{a_j}) and it is defined as follows:

$$U_{a_j} = (1 - \prod_{S_i \rightarrow T_j} (1 - e_{ij})) \tag{1}$$

After running NUM, each source is assigned a specific rate at which it can send data to the task's sink. As described in [9], the *rate utility* u_{r_i} of sensor S_i sending at rate r_i must be concave for NUM to find the optimal solution. In this paper we assume that it is the following function:

$$u_{r_i} = \sqrt[3]{{r_i}^2} \tag{2}$$

The rate utility of task T_j is equal to the sum of the rate utilities of its sources.

To normalize the value of the rate utility we assume that the best rate utility that a single sensor can achieve is 1200 which corresponds to a bit rate of 41.5Kbps. If a sensor achieves this rate utility then the normalized utility is 1. If the rate utility is 600 then the normalized rate utility is 0.5. When N sensors are assigned to a task then best rate utility becomes 1200 $\times N$.

If the assignment utility is U_{a_j} and the normalized rate utility is U'_{r_j} , then the combined utility of task T_j is:

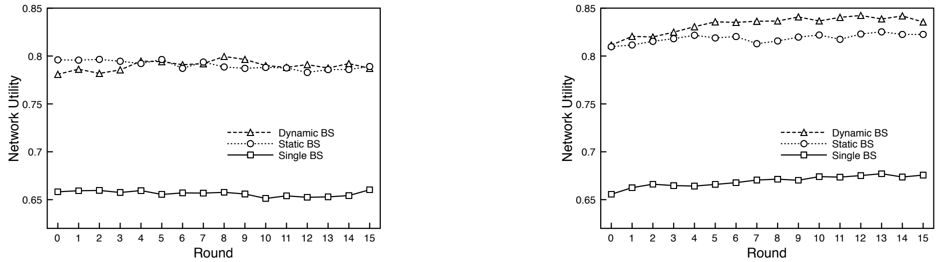
$$U_j = U_{r_j} \times U'_{a_j} \tag{3}$$

The average performance of a task is used as a measure for the utility of the network. The figures show the results averaged over 15 runs. We run the two algorithms for 15 rounds disregarding the termination point to study their behavior.

B. Simulation Results

Figure 1 shows the results when we limit the number of sensors that can be assigned to a task to one. The figures shows the performance as the algorithms progress through 15 rounds (the two plots use the same scale for easier comparison). We find that the curves for the three base selection schemes of G-D stay nearly flat with the static base station (**BS**) and dynamic BS closely aligned. The curves for NBI follow the same trend but with slightly higher network utility. This is because with only one sensor assigned per task the assignment utility matters more than the rate utility. When one sensor is chosen then its detection probability is the assignment utility of the task. Because the G-D algorithm takes interferers into account it penalizes sensors on congested routes and chooses others that are not. The other sensors, however, may provide lower utility that decision can lower the overall utility.

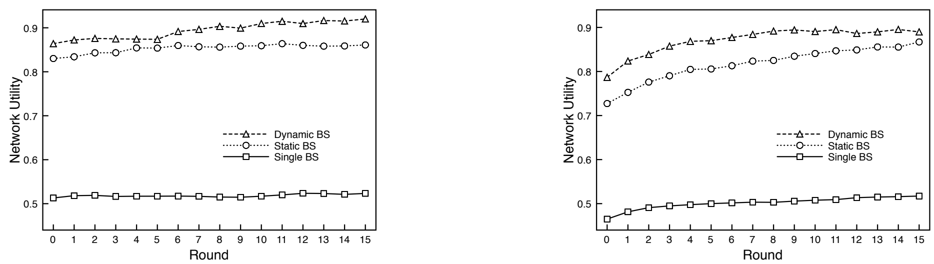
Figure 2 shows the results when we limit the number of sensors that can be assigned to a task to four. The figure shows the performance as the algorithms progress through 15 rounds. As with the one-to-one case, we find that the curves for the three base selection schemes of the G-D algorithm stay nearly flat. For NBI, we see an increase in network utility as we progress through the 15 rounds. From the initial assignment (round 0 in the figures), we note an improvement of 5% to 14% depending on the BS selection used.



(a) Greedy-Divide (G-D) Algorithm

(b) NUM-based Iterative (NBI) Algorithm

Fig. 1. Results for one-to-one assignment with 16 tasks



(a) Greedy-Divide (G-D) Algorithm

(b) NUM-based Iterative (NBI) Algorithm

Fig. 2. Results for many-to-one assignment with 4 tasks each assigned 4 sensors

G-D on the other hand, starts higher and does not change significantly through the rounds. This suggests that because G-D takes the interferers into account when making the initial assignment, it is able to eliminate most of the interference that is faced by NBI. This allows the algorithm to converge faster than NBI.

As for the BS selection schemes, we note that the dynamic BS selection gives the best results. This is to be expected since it chooses the BS with the least number of interferers along the route which leads to higher data rates. Static BS selection comes as a close second. By selecting the BS that is closest in terms of number of hops, we are able to distribute the load and eliminate cross traffic (i.e. traffic going from one side of the network to the other). This again leads to fewer interferers and hence higher data rates. Distant third is the single BS scheme. It has the lowest performance since it forces data from all sources of a task to go to the same BS which can be located any where in the network and not necessarily close to the task's location. This does not only cause cross traffic but also contention at the last hop to the task's BS which is shared by all sources. Ultimately, this leads to very low rate utility.

In [9] it was found that running NUM requires about 200 seconds per round. This means that for 10 rounds it will take NBI over 3 minutes to run. Because G-D does not run NUM in each round it can complete much faster. We estimate that 10 rounds of G-D will take an order of magnitude shorter time which will make it more desirable in real-time environments.

VIII. Conclusion

In this paper we proposed algorithms to solve the combination of two previous problems, namely sensor-task assignment and rate utility optimization. We found through simulation that we can improve the network utility from the initial assignment by 5% to 14%. We also found that both algorithms ultimately achieve nearly the same results but the Greedy-Divide algorithm can converge much faster than the NUM-based Iterative algorithm.

Acknowledgements This research was sponsored by US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the US Government, the UK Ministry of Defence, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] J. Byers and G. Nasser, "Utility-based decision-making in wireless sensor networks," in *Proceedings of MobiHoc 2000*.

[2] M. Perillo and W. Heinzelman, "Optimal sensor management under energy and reliability constraints," in *Proceedings of the IEEE Conference on Wireless Communications and Networking*, March 2003.

[3] K. Shih, Y. Chen, C. Chiang, and B. Liu, "A distributed active sensor selection scheme for wireless sensor networks," in *Proceedings of the IEEE Symposium on Computers and Communications*, June 2006.

[4] A. Bar-Noy, T. Brown, M. Johnson, T. La Porta, O. Liu, and H. Rowaihy, "Assigning sensors to missions with demands," in *ALGOSENSORS 2007*.

[5] H. Rowaihy, M. Johnson, A. Bar-Noy, T. Brown, and T. La Porta, "Assigning sensors to competing missions," in *Globecom '08*.

[6] M. Johnson, H. Rowaihy, D. Pizzocaro, A. Bar-Noy, S. Chalmers, T. La Porta, and A. Preece, "Frugal sensor assignment," in *DCOSS '08*.

[7] F.Kelly, *Charging and Rate Control for Elastic Traffic*.

[8] F.Kelly, A. Maulloo, and D. Tan, *Rate control for Communication Networks: Shadow Prices, Proportional Fairness and Stability*.

[9] S. Eswaran, A. Misra, and T. L. Porta, "Utility-based adaptation in mission-oriented wsn," in *Proc. of IEEE SECON, June 2008.*

[10] S. Eswaran, M. Johnson, A. Misra, and T. L. Porta, "Distributed utility-based rate adaptation for prioritized missions," Pennsylvania State University, Tech. Rep. TR NAS-TR-0090-2008, 2008.

[11] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Colmpleteness*. Freeman, 1979.