Game Theoretic Multi-Agent Systems Scheduler for Parallel Machines

Elisha T. O. Opiyo, Erick Ayienga, Katherine Getao, & William Okello-Odongo[*]
School of Computing and Informatics,
University of Nairobi

Bernard Manderick, & Ann Nowé
Computational Modeling Laboratory,
Free University of Brussels

This paper considers the scheduling of *n* independent jobs on *m* non-identical machines using the ideas from game theory and multi-agent systems. The values of n and m are fixed at 30 and 6 respectively giving a schedule space with a dimension of approximately 10^{23} schedules. The agents are used to represent the jobs and they select machines on which the jobs should be processed, resulting into schedules. The schedules that are generated are evaluated using the *makespan* which is the total time taken for all the jobs to be processed. The *makespan* of the schedules that are generated vary when the agents that represent the jobs change the way they make their selection decisions. The agent selection policies that are investigated in this paper include pure random choice, potential game strategy and dispersion game strategy. The results that are obtained show that the random choice strategy and the potential game strategy generate the empirical best schedules by chance. The dispersion game strategy however is shown to converge very quickly to a stable schedule type whose best *makespan* value is between 3.1 to 3.4 times larger than the empirical best schedule. The main contributions in this paper include generating schedules in a concrete schedule space using ideas from game theory and multi-agent systems and the results that are obtained.

Categories and Subject Descriptors: [Artificial Intelligence Applications]:- *Multi-Agent Systems*;
General Terms: Agents, Scheduling, Parallel Machines
Additional Key Words and Phrases: Game theory*;* grid computers

IJCIR Reference Format:

Elisha T. O. Opiyo, Erick Ayienga, Katherine Getao, William Okello-Odongo, Bernard Manderick, and Ann Nowé. Game Theoretic Multi-Agent Systems Scheduler for Parallel Machines. International Journal of Computing and ICT Research, Special Issue Vol. 1, No. 1, pp. 21-27. http://www.ijcir.org/Special-Issuevolume1-number1/article 3.pdf.

1. INTRODUCTION

---

[*] Author's Address: Elisha T. O. Opiyo, Erick Ayienga, Katherine Getao, and William Okello-Odongo, School of Computing and Informatics, University of Nairobi; Bernard Manderick, and Ann Nowé, Computational Modeling Laboratory, Free University of Brussels

International Journal of Computing and ICT Research, ISSN 1818-1139 (Print), ISSN 1996-1065 (Online),), Special Issue Vol.1, No.1 pp. 21-27, October 2008.

Scheduling of a given number, *n* of independent jobs on a given number, *m* of non identical parallel machines is an area that has attracted much interest for several years especially from Operations Research [Graham et al. 1977; Sing & Sheik 1998; Cicirello 2003; Nowicki and Smatnicki 2005; T'Kind and Billaut 2006; Johannes 2006; Trietsch and Baker 2008]. The scheduling problem has also attracted the interest of researchers from Artificial Intelligence [Jurgen 1998; Jennings and Jackson 1995; Sycara et al. 1995; Opiyo et al. 2005; Opiyo et al. 2006; Opiyo et al. 2007]. Scheduling independent jobs on parallel non identical machines seek algorithms that optimize, through minimization, such factors as the total time it takes for all the jobs to be completed, also called the *makespan* [Ibarra and Kim 1977]. The scheduling problem has been well structured for the job shop related applications. Graham et al. [1977] introduced a three component format for specifying the problems in which the machine properties are specified followed by the constraints then the optimization details. In that format the general class of the scheduling problem handled in this paper is $Q||C_{max}$, denoting uniform machines, with no constraints specified and minimizing the total completion time for al the jobs. In this paper the values of n and m are fixed at 30 and 6 respectively giving a schedule space with a dimension of approximately 10^{23} schedules. The schedules are generated using ideas from Game theory and Multiagent systems. This differs from other approaches based on Operations Research (OR). The issue with the OR approaches is that most solutions are limited to each class of the scheduling problem that is solved. This makes it necessary to seek the invention of algorithms or heuristics for different problem classes. For example algorithms for 1||Cmax are not guaranteed to solve the 3||Cmax or the Q||Cmax problems. The agent-based approaches are different. The schedules are generated according to the agent behaviour. This associates the qualities of schedules that are produced with the behaviour of the agents. This shifts the burden of the scheduling problem from the invention of algorithms to determining the agent behaviour that would lead to good schedules. The main advantage of using the agent-based approach is that in the extreme case that the problem class is unfamiliar the agents can learn the behaviour that leads to good schedules on their own, see Galstyan et al. [2005]. The scalability problem is also less restricting with the agent based approach. The agent based approach is also more appropriate for the emerging distributed systems environments such as the Internet, cellular networks and grid computing [Foster et al. 2001; Araujo et al. 2007; Berman et al. 2003].

1.1 The scheduling problem as a multiagent systems game

In this section the way that the scheduling process is modeled as the agent game is outlined. The agents are entities that can sense and react to changes in their environments autonomously. The multiagent system is a system that consists of the agents [Wooldrige 2002]. Game theory is the study of interactions in contexts where the participants make the choices that are conflicting. A game is a structure that consists of a set of the agents, a set of the agent actions or choices and a set of the agent payoffs associated with their actions [Wooldrige 2002; Shoham and Leyton-Brown 2008]. The scheduling task under consideration involves assigning *n* independent jobs that are to be processed on *m* non identical machines so that the total processing time for all the jobs is minimized. A game can be formulated in which the agents represent the jobs, the machines or both [Porter 2004; Heydenreich et al. 2006; Angel et al. 2006]. In this work the agents represent the jobs. The agents act by choosing the machines on which the jobs that they represent are to be processed. The agent payoffs are based on the *makespan*. A schedule with a smaller *makespan* is the one that is more preferred. The makespan is however, a group payoff rather than an individual payoff. The types of games that are of interest in this paper include dispersion games [Trond et al. 2002], potential games [Monderer and Shapley 1996], and random choice games. Random choice games are those in which the agents make choices at random without considering any other matters.

Dispersion games are those in which the agents win positive payoffs when they choose distinct actions [Trond et al. 2002; Yuri et al. 1997]. The agents therefore prefer to be more dispersed over actions in that they choose deferent actions than those chosen by other agents. For example, setting up new businesses in areas where there are no similar businesses and choosing to drive on streets with low traffic, are some of the activities that can be modeled by dispersion games.

Potential games are those in which the incentive of all players to change their strategy is expressed in one global function called the potential function [Monderer and Shapley 1996; Sandholm 2001]. The progressive actions of the participants lead to a stable state. The use of taxes or public charges to influence the decisions of people is a form of potential game.

The schedules that are generated are evaluated using the *makespan*. The schedule with the least makespan is the better one. The behaviour of the agents is pre-determined and in each case the resulting schedules are examined. The agents represent the jobs and they select the machines on which the jobs should be processed. The way that the agents select the jobs constitute their behaviour. These different selection schemes give rise to different classes of schedules based on the *makespan*. The qualities of these schedules are compared. The broad objective of this work is to try to identify the agent selection scheme that would yield good schedules. The specific objective is to find out how the quality of the schedules is affected by the random choice games, the potential games and the dispersion games.

1.2 Game theoretic multi-agent systems scheduler

In this section the scheduler that relies on the ideas from game theory and multi-agent systems is presented. The core of the algorithm that is used is also given. There are 30 jobs each represented by an agent and there are 6 machines. A typical best schedule that was generated is given in Figure 1.

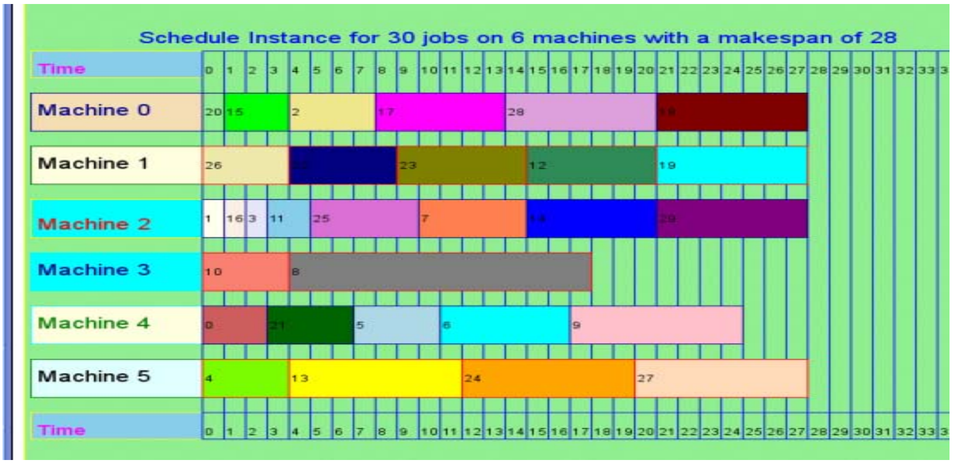

Figure 1 : Typical best schedule with a makespan of 28

The agents are labeled 0 – 29 and the machines are labeled 0 – 5. The machines are shown to the left. Each machine has the jobs scheduled on it labeled 0 – 29 on its right. The core of the allocation algorithm is outlined as follows:

```
Initialization: AgentList(0-29) each with processing time, MachineList(0-5) each with a speed
While (rounds < some pre-determined value- 100,000: 1,000,000 )
        Agents select machines ( random || potential || dispersion game strategy)
        Shortest processing time (SPT) heuristic is used to arrange agents on local machines
        Note schedule statistics; adjust rounds
EndWhile
```

The remaining sections of this paper include related work where similar research is highlighted; the experimental set up that outlines the simulation experiments and the results that were found; the discussion of the findings, and the conclusion.

2. RELATED WORK

This work was motivated by the work of Heydenreich et al. [2006] and Porter [2004] in their work that is part of the broader area of algorithmic game theory [Marios et al. 2006; Nisan et al. 2007; Shoham and Leyton-Brown 2008]. Porter's model had *n* jobs that were represented by the agents, a centre that applied a scheduling mechanism and *one* machine processor. This was of the type $1||C_{max}$ using the notation of Graham et al. [1977]. This model was extended by Heydenreich et al. [2006] to the general type $P||C_{max}$ in which several jobs run on several *identical* parallel machines. The jobs are represented by the agents. The agents select the machines on which they want the jobs to be processed. This work differs from their work in that the problem class that is considered is in this work is of the type $Q||C_{max}$, where the machines are not identical. It also differs in the approach in that these researches used analytical techniques to investigate the loss of efficiency from a central to a distributed environment using the price of anarchy. This work on the other hand conducts investigations in a concrete space, uses an empirical approach, and investigates the quality of schedules that can be obtained by varying the agent behaviour.

3. THE EXPERIMENT AND THE RESULTS

This section gives an outline of the way that the experiments were conducted. The aim of the experiment was to determine the agent behaviour that generates the schedules that are good, that is with minimum *makespan*. The important properties of jobs include the release time, the deadline, the processing time and the weight. The important machine property is its speed. There were 30 jobs labeled 0 to 29. There were 6 machines labeled 0 to 5. The machine speeds and the job processing times were fixed using integer values. It was assumed that the scheduling process begins when all the jobs are available. The release time was therefore fixed to 0 in all cases. It was also assumed that the jobs were equally weighted. The schedules were produced in rounds. One round gave one schedule. After each round, the schedule, the round of the schedule and the *makespan* of the schedule were noted. If the schedule turned out to be better than the best schedule then its details were remembered, replacing those of the incumbent best schedule. If the schedule turned out to be the worse than the worst schedule then its details replaced those of the worst incumbent schedule. The rounds ranged from 100, 000 to 1 million.

The empirical best *makespan* was 28 while the empirical worst *makespan* was 400. These values only hold for the fixed integer values that were used for the processing times of the jobs and the speeds of the machines. The results are summarized in Table 1, and Figure 2.

Table 1: Summary of the results

Attribute	Random Choice	Potential Game	Dispersion Game
Best makespan	28 - 30	28 – 30	30 -60
Worst makespan	350 - 480	270 – 380	100 - 220
Effort for best-	65%	75%	Under 5 %
Effort for worst -	15%	10%	Under 5%
Schedule quality-	Just below 8 %	7% - 10 %	8% constant
Quality of best	98%	82 %	35 %
Quality of worst	5%	5%	10%
Probability distr.	90 % in 158 - 167	90% in 118 – 127	99 % in 88-97

Legend: best- :best makespan; worst- :worst makespan; quality-: quality index
distr. :distribution; 158-167, 118-127, 88-97: intervals for makespan

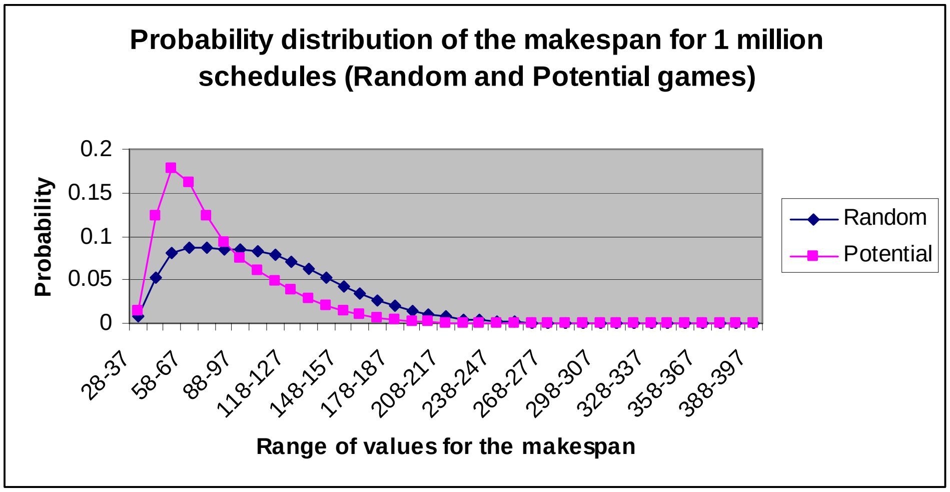

Figure 2: Distribution of the *makespan* based on Random Choice and Dispersion game strategies

4. DISCUSSION

The results on the best *makespan* obtainable using various agent selection schemes indicate that only the random choice and the potential game strategies give a *makespan* equal to or close to 28. The *makespan* of 28 is the empirical best *makespan* for the given integer values of the job processing times and the machine speeds [Table 1]. The results from dispersion game strategy are not as good since the best *makespan* lies between 30 and 60. This is explained by the fact that the random effect that is present in both random choice and potential games causes the free exploration of the schedule space and makes it possible for them to stumble on schedules with the best *makespan*. The dispersion game strategy scheme on the other hand converges much faster to the load balanced configurations that restrict the *makespan* to a similar range of 88 – 97 [Table 1]. This makes the best and the worst schedules to occur around this range for the dispersion game strategy. On whether there is some predictable amount of search effort required to get the best schedule the results indicate that no reliable predictable rule is immediately evident for random choice and potential game strategies. From the results it is only indicative that more rounds are needed to guarantee getting a good schedule when using the random choice or the potential game selection policies. The dispersion game gives a very different result. The number of rounds, for getting both the best and the worst schedules is extremely low. This is explained by the convergence to a balanced load on all machines that occurs very fast mostly by the third schedule generated or third round. The results indicate that highest quality and lowest quality schedules are generated using the random and potential game strategies. Dispersion game strategy, however, produces schedules that are moderate, though between 3.1 and 3.4 larger than the empirical best schedule. These schedules are not so good and not so bad. The distribution of schedules in the space was also investigated [Figure 2 and Table 1]. The random choice strategy scheme gives insight into the possible schedule distribution in the schedule space based on the quality of the *makespan*. This is because the random choice strategy generates schedules at random and independently and therefore draws random samples from the schedule space. The distributions obtained from the potential game strategy and dispersion game strategy only depict the way the agent behaviour affects quality schedules. The potential game is designed such that the agents will tend to repeat the actions that previously led to the best schedules. This is visible in Figure 2, where the displacement is to the left, or towards the good schedules, those with smaller *makespan*. The dispersion game however led to the schedules with the *makespan* mostly in the range of 88 – 97 [Table 1].

5. CONCLUSION

This work has taken an empirical approach to handling the scheduling problem involving independent jobs on parallel non identical machines. The long term objective is to find a way of producing schedules that are good based on some measure and in this case, the *makespan*. It has been demonstrated that the ideas from game theory and multi-agent systems can be used to generate schedules in a concrete space. It has also been shown that the quality of schedules is affected by the agent behaviour as it selects a machine. It has also been demonstrated that there is a schedule space in which the performance of load balanced solutions is consistently 3.1 to 3.4 times worse than the empirical best. In real life terms the scheduling scenarios occur in more complex contexts than the 30 jobs and 6 machines scenario. The schedule spaces in such scenarios are much larger. It means that there is much room, and by some chance, to operate with schedules that are close to the best and plenty of room, also by some chance, to operate with schedules that are bad. The findings in this work point to further investigations on the desired agent behaviour since none of the selection schemes seem satisfactory. The dispersion game converges very quickly to a solution but such as solution can be far from the empirical best schedules. This work has investigated the quality of schedules where some given integer values of the job processing times and the machine speeds are used, however, it may be extended to other selected integer values. It may also be extended to investigate contexts in which the jobs and the machines are dynamic. These are left as the issues for further work. Another area for further work is to investigate if it is necessary to use any heuristics at all where the agents are used to produce schedules as they play the allocation game for the scheduling problems of the type $Q||C_{max}$.

6. ACKNOWLEDGMENTS

Our work is supported by the Flemish Inter-University Council: Institutional University Cooperation Programme (VLIR-IUC) of Belgium. We thank them for this support. We are also grateful to the VLIR

Secretariat at the University of Nairobi, and the Free University of Brussels (VUB), for their support. We also appreciate the contributions of the anonymous reviewers in improving this paper.

7. REFERENCES

ANGEL ERIC, EVRIPIDIS BAMPIS, FANNY PASCUAL. 2006. *Algorithmic Game Theory and Scheduling.* IBISC, University of Every, France. CIRM, May 2006. vs **..** www.cirm.univ-rs.fr/videos/2006/exposes/14/Monday%20Afternoon/Bampis.ppt - last visited February, 2008.

ARAUJO, RICARDO MATSUMURA DE; LAMB, LUIS C. 2007. An information theoretic analysis of memory bounds in a distributed resource allocation mechanism**.** In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence* (IJCAI 2007), 2007, Hyderabad, India.

BERMAN FRAN, FOX GEOFFREY C., HEY ANTHONY J. G. 2003. Editors. *Grid computing: making the global infrastructure a reality*. Wiley publishers.

CICIRELLO VINCENT A. 2003. Weighted tardiness scheduling with sequence-dependent setups: A benchmark library. *Technical report, Intelligent Coordination and Logistics Laboratory*, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, February 2003.

FOSTER IAN, CARL KESSELMAN, STEVEN TUECKE. 2001. *The Anatomy of the grid: enabling scalable virtual organizations*, reprint in Berman et al. (2003) pp. 171 – 191.

GALSTYAN ARAM, KARL CZAJKOWSKI, AND KRISTINA LERMAN. 2005. Resource Allocation in the Grid with Learning Agents. In the *Journal of Grid Computing,* Volume 3, pages 91-100, Springer 2005.

GRAHAM R. L, E. L. LAWLER, J. K. LENSTRA, A.H. G. RINNODY KAN. 1977. Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. Stichting Mathematisch Centrum. Mathematische Besliskunde; BW 82/77.

HEYDENREICH BIRGIT, RUDOLF MÜLLER, MARC UETZ. 2006. Decentralization and Mechanism Design for Online Machine Scheduling, Algorithm Theory- SWAT 2006, Volume 4059, *Lecture Notes in Computer Science*, pp 136-147. Springer.

HEYDENREICH BIRGIT & MÜLLER RUDOLF & UETZ MARC. 2006b. Games and Mechanism Design in Machine Scheduling – An Introduction, *Technical Report 022*, Maastricht : METEOR, Maastricht Research School of Economics of Technology and Organization, 2006.

IBARRA OSCAR H., CHUL E. KIM. 1977. Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors. In *Journal of the Association for Computing Machinery*, Vol 24, No. 2, April 1977, pp 280-289.

JENNINGS N. R., A. J. JACKSON. 1995. Agent-based Meeting Scheduling: A Design and Implementation. *IEE Electronics Letters*, Vol. 31, No. 5, The Institution of Electrical Engineering, pp. 350--352, 1995.

JOHANNES BERIT. 2006. Scheduling parallel jobs to minimize the makespan. In Journal of Scheduling, 9 pp. 433 – 452, Springer.

JURGEN DORN. 1998. *Towards reusable intelligent scheduling software. In Knowledge-Based Systems*, pp. 101-112, 1999, citeseer.ist.psu.edu/456393.html.

MARIOS MAVRONICOLAS, VICKY PAPADOPOULOU, PAUL SPIRAKIS. 2006. Algorithmic Game Theory and Applications. *Wiley-Interscience*, John Wiley & Sons publication.

MONDERER DOV, LLOYD S. SHAPLEY. 1996. Potential Games. In *Games And Economic Behavior* 14, 124–143 (1996) Article No. 0044.

NISAN NOAM, TIM ROUGHGARDEN, ÉVA TARDOS, VIJAY V. VIZIRANI. 2007. *Algorithmic Game Theory*. Cambridge University Press, 2007.

OPIYO ELISHA T. O., ERICK AYIENGA, KATHERINE GETAO, BERNARD MANDERICK, OKELLO-ODONGO, ANN NOWÉ. 2005. Computing research challenges and opportunities with grid computers. In proceedings of *The first International workshop on sustainable research excellence in computing for developing countries*. 9-12 August 2005, Makerere, Uganda.

NOWICKI EUGENIUCZ, CZESLAW SMATNICKI. 2005. An Advanced Tabu Search Algorithm for the Job Shop Problem. In *Journal of Scheduling*, 8, pp. 145-159, 2005.

OPIYO ELISHA T. O., ERICK AYIENGA, KATHERINE GETAO, BERNARD MANDERICK, OKELLO-ODONGO, ANN NOWÉ. 2006. Multi-agent systems scheduler and open grid computing standards. In *Proceedings of The secondt International workshop on sustainable research excellence in computing for developing countries*, 9-12 August 2006, Makerere, Uganda.

OPIYO ELISHA T. O., ERICK AYIENGA, KATHERINE GETAO, BERNARD MANDERICK, OKELLO-ODONGO, ANN NOWÉ. 2007. Modeling a Multi-Agent Systems Scheduler for Grid Computing Using the Gaia Methodology. In *1st International Conference in Computer Science and Informatics*, 5th – 7th February 2007, Nairobi, Kenya, School of Computing & Informatics, University of Nairobi, Kenya.

PORTER RYAN. 2004. Mechanism Design for Online Real-Time Scheduling. Proc. 5th *ACM Conf. Electronic Commerce*, ACM Press, 2004, 61-70.

SANDHOLM W. H. 2001. Potential Games with Continuous Player Sets. In Journal of Economic Theory, Volume 97, pp. 81 – 108, 2000.

SHOHAM YOAV, KEVIN LEYTON-BROWN. 2008. *Multiagent Systems; Algorithmic, Game-Theoretic and Logical Foundations.* Cambridge University Press, 2008.

SINGH JAIN ANANT, MEERAN SHEIK. 1998. A state-of-the-art review of job-shop scheduling techniques. *A Technical report, Department of Applied Physics, Electronic and Mechanical Engineering,* University of Dundee, Dundee, Scotland, 1998.", citeseer.ist.psu.edu/jain98stateart.html. Last visited January 2008.

SYCARA KATIA, DAJUN ZENG, KAZUO MIYASHITA. 1995. Using CBR to acquire user scheduling preferences. In *Proceedings of the Eleventh IEEE Conference on Artificial Intelligence Applications*, 1995. Www.cs.cmu.edu/~sycara/cabins-papers/caia95-final.ps.gz

T'KINDT VINCENT, JAN-CHARLES BILLAUT. 2006. *Multi-criteria scheduling: Theory, models and algorithms.* Springer-Verlag.

TRIETSCH DAN, KENNETH R. BAKER. 2008. Minimizing the number of tardy jobs with stochastically ordered processing times. In *Journal of Scheduling*, 11, pp. 71 – 73, Springer.

TROND GRENAGER, ROB POWERS, YOAV SHOHAM. 2002 Dispersion Games: General Definitions and Some Specific Results. In *proc. AAAI02*, pages 398-403. AAAI Press.

WOOLDRIDGE MICHAEL. 2002. *An introduction to multiagent systems*. John Willey & Sons, Ltd, 2002.

YURI M. ERMOLIEV, SJUR DIDRIK FLAM. 1997. Learning in Potential Games. *INTERIM REPORT IR-97-022 / June, International Institute for Applied Systems Analysis* _ A-2361 Laxenburg _ Austria.