Paving the Royal Road for Complex Systems: On the Influence of Memory on Adaptivity

Christian Hinrichs[1], Sebastian Lehnhoff[2], and Michael Sonnenschein[1]

[1] University of Oldenburg, Germany,
christian.hinrichs@uni-oldenburg.de
sonnenschein@informatik.uni-oldenburg.de
[2] OFFIS Institute for Information Technology, Oldenburg, Germany,
sebastian.lehnhoff@offis.de

Abstract. The "Royal Road" objective function was proposed by J. H. Holland in 1993 as a very hard benchmark problem for evolutionary algorithms. Generally, it belongs to the class of combinatorial optimization problems. In our work, we solve the problem in a distributed way by assigning each decision variable to an autonomous agent. The resulting multi-agent system "COHDA" forms a self-organizing complex system, where the global solution emerges from local interactions. By applying the XOR instance generator introduced by S. Yang in 2003, we are able to pertubate the system during runtime by modifying the objective function. This allows us to examine the robustness of COHDA against dynamic objectives. Here, we focus on the influence of runtime memory, which comprises the beliefs of each agent, on the adaptivity capabilities of the agents after an occured pertubation. We show that the final fitness values produced by the system do not suffer from a dynamic objective function, and are not influenced by the availability of an agents' runtime memory. The time needed by the system to adapt to such a pertubation, however, significantly increases if the agents' beliefs are being distorted. We conclude that, in terms of solution quality, COHDA is very robust against dynamic objective functions. With respect to adaptation speed, the heuristic benefits from the availability of runtime memory.

Keywords: Combinatorial Optimization, Self-Organization, Cooperation, Multi-Agent System

1 Introduction

In a general way, an optimization problem can be characterized as follows [1]: Given a set $\mathcal{S}$ of feasible solutions and an objective function $f : \mathcal{S} \to \mathbb{R}$, one tries to find those elements $s \in \mathcal{S}$ which maximize (or minimize) f. Typically, the elements in $\mathcal{S}$ are tuples comprising values for a set of decision variables $v_0 \dots v_n$. If the feasible values for these variables are discrete rather than continuous, this is called a combinatorial optimization problem. Instances of this family of problems are usually hard to solve, since their structure is less exploitable than in the continuous case. Even more, many of such problems are

A. Pelster and G. Wunner (Eds.): International Symposium Selforganization in Complex Systems: The Past, Present, and Future of Synergetics, DOI: 10.1007/978-3-319-27635-9_21, pp. 313–318.

computationally intractable, so that exact solution methods are not appropriate [2]. Hence, a number of approaches have been proposed, which aim at finding a *quite good* solution in a *reasonably short* amount of time, but without guaranteeing any particular solution quality with respect to f. These approaches are called heuristics.

A special form of combinatorial optimization problem arises, if each variable $v_0 \dots v_n$ is controlled by an autonomous decision maker $a_0 \dots a_n$, respectively. Example applications for this kind of problem include distributed resource allocation, logistics and decentralized energy management. In such distributed systems, the decision makers (in the following simply denoted as agents) have to *coordinate* their decisions in order to jointly optimize the objective function f. Hence, when designing efficient solution strategies for such systems, not only computational complexity as well as memory complexity, but also communication complexity (information exchange between agents) has to be regarded.

Recently, the "COHDA" heuristic has been proposed, which utilizes a self-organization strategy in order to solve distributed combinatorial optimization problems efficiently in a completely decentralized and asynchronous way, and thus forms a nonlinear complex system, where the global solution emerges from local interactions [3, 4]. In the contribution at hand, we focus on the adaptivity capabilities of COHDA with respect to the amount of runtime memory of the underlying agents, using the example of a dynamic variant of the "Royal Road" benchmark problem.

2 The Dynamic Royal Road Benchmark Problem

The "Royal Road" objective function was proposed by J. H. Holland in 1993 as a very hard combinatorial optimization problem. The function takes a tuple $s = v_0 \dots v_{(2^k)\cdot(b+g)-1}$ of binary values as input, such that $v_i \in \{0, 1\}$, and produces a fitness value $f(s) \in \mathbb{R}$. The goal is to maximize f. The numbers k, b, g are predefined integer parameters, such that the tuple s comprises 2^k contiguous *regions*, each containing b values forming a *block*, and g values forming a *gap*. The evaluation criteria of f are designed such that each region contributes a higher fitness to the resulting global fitness value, the more 1's are contained in its block part, but only up to a certain threshold $m^* < b$. For amounts of 1's in the range $[m^* + 1, b - 1]$, a region would yield a negative fitness, and thus contribute a penalty to the global fitness value of the function. Finally, for a block full of 1's (this is then called a complete block), a region would again yield a large positive value. Furthermore, series of regions with complete blocks each would produce extra bonus values to the global fitness. In this whole fitness evaluation, the gaps do not have any influence on the resulting value. Hence, the optimal global fitness value will be reached if every block is "complete", and the worst solution (with a negative global fitness value) will be produced if every block contains exactly $b - 1$ ones.

In an iterative search process, most optimization algorithms would add more and more 1's to each block, until eventually each block comprises m^* ones. Since

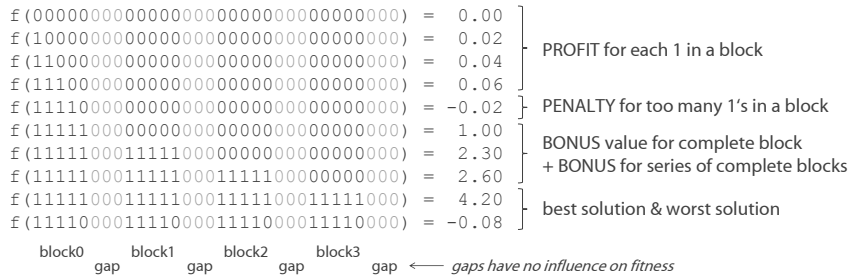

Fig. 1. Example of a "Royal Road" objective function ($k = 2$, $b = 5$, $g = 3$, $m^* = 3$).

adding more 1's would introduce penalties and thus decrease the global fitness value, this poses a local optimum. But in order to reach the global optimum, an optimization procedure would have to pass the areas with very low fitness values in the solution landscape. Because of this property, the function is called *deceptive*. Figure 1 shows an example configuration. A more detailed description of Holland's "Royal Road" function can be found in [5].

Since the contribution at hand targets the adaptivity capabilities of the COHDA heuristic, the "Royal Road" objective function had to be converted to a dynamic variant. This can be done using the XOR instance generator introduced by S. Yang in 2003 [6]. For this purpose, a randomly chosen binary tuple T with a predefined Hamming weight $\mathrm{HW}(T)$ is chosen, and the objective function f is replaced by a function g, which is defined as $g(s) = f(s \otimes T)$. Here, the operator $\otimes$ is a component-wise XOR operation, thus changing the "meaning" of every value in s, where the corresponding value in T equals to 1. This transforms the solution landscape of the objective function without affecting its basic structure (i.e. number of optima). The Hamming weight determines the severity of the transformation: $\mathrm{sev}(T) = \frac{\mathrm{HW}(T)}{\mathrm{len}(T)}$. By applying such randomly chosen transformations at arbitrary time steps t during the optimization procedure, this allows us to analyze the behavior of the COHDA heuristic in a perturbed system.

3 COHDA

As stated in the introduction, COHDA is a heuristic for solving distributed combinatorial optimization problems. Basically, each decision variable of a given problem is controlled by an autonomous decision maker, implemented as a software agent in a multi-agent system. The agents are allowed to communicate through a communication network, whose topology forms a partially connected, undirected graph. Figure 2 shows an example of such a system. In COHDA, a global solution emerges from local interaction between agents. Each agent follows the same simple behavioral rules: 1) Upon an incoming information from a neighbor, the local knowledge base of the agent is updated. 2) Afterwards, the agent chooses the value for its controlled decision variable, that suites the

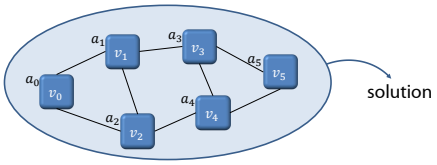

Fig. 2. Visualization of an exemplary communication topology used by COHDA.

currently believed system state the best. 3) The agent publishes its local knowledge base to the neighborhood. Following these three steps, the system will first asynchronously explore the solution space, before converging to the best solution found by an agent. More details on the heuristic can be found in [3, 4].

4 Evaluation

Our evaluation focuses on the adaptivity capabilities of COHDA in a dynamic, perturbed environment. The pertubations are modeled by a dynamic objective function as described in Sect. 2. In more detail, we analyzed the influence of an agent's runtime memory on the efficiency of the heuristic to converge after an occured pertubation. A simulation study has been performed using a royal road function with $k = 3$, $b = 5$, $g = 3$, $m^* = 3$. We compared different pertubation severities $\text{sev}(T) \in \{\text{None}, 0.0, 0.1, 0.2, 0.5\}$ against a number of configurations of runtime memory:

- **No memory.** Whenever a pertubation occured, the knowledge bases of all agents are completely erased, and each agent is re-initialized with a random value for its decision value.
- **Limited memory.** Same as above, except that the currently selected value for the decision variables are kept.
- **Full memory.** Knowledge bases are kept intact upon system pertubation.

Additionally, we included for reference an evolutionary algorithm with one parent, offspring of size 1 and adaptive mutation rate ((1+1)-ES, c.f. [7]). Each configuration was simulated 100 times, the results are summarized in Fig. 3, showing mean values and standard deviations. The upper chart presents the fitness of the solution produced by the algorithms, normalized to $[0, 1]$, whereas the lower chart depicts the number of iterations until convergence, as a measure of the time needed to converge. Concerning fitness, the COHDA heuristic produces significantly better results than the reference algorithm in all cases. Also, COHDA is unaffected by objective pertubations during runtime. With respect to the number of iterations until convergence, however, the results show that the heuristic benefits from the existence of runtime memory when a pertubation occurs, especially with low pertubation severities. Obviously, the agents make use of their existing memory in order to adapt to a changing objective function as fast as possible.

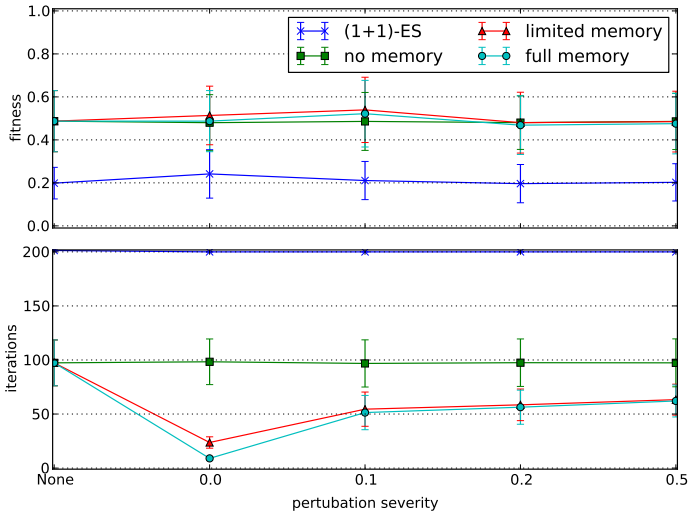

Fig. 3. Evaluation results: Pertubation severities compared against different levels of runtime memory, (1+1)-ES included for reference.

5 Application

The primary use case of the COHDA heuristic lies in the domain of decentralized energy management systems, as it is envisioned for example in the Smart Nord research project [8]. Here, coalitions of intelligent generators, loads and storages are formed in order to provide active power as well as ancillary services in the power grid. Within this research project, the goal of a coalition is to provide a *product* of either active power or ancillary services. This product can then be placed at a market (see Fig. 4). However, in the case of an active power product, the selection of an active power profile for each agent in the coalition, in order to jointly produce the desired product, forms a combinatorial optimization problem as described in the introduction. The COHDA heuristic can be used to solve this problem efficiently in a distributed way.

However, the COHDA heuristic relies on cooperative agents in principle. In our future work, we will study the influence of non-cooperative (i.e. self-interested) agents on COHDA. How many private constraints (with respect to an agent's interests) can the heuristic cope with? Up to which amount of non-cooperative agents is the heuristic still effective?

6 Conclusion

Our world gets more and more connected, evolving to an *internet of things*, where the interconnected entities get smarter every day. In our research, we focus on decentralized energy management systems. We believe that in such

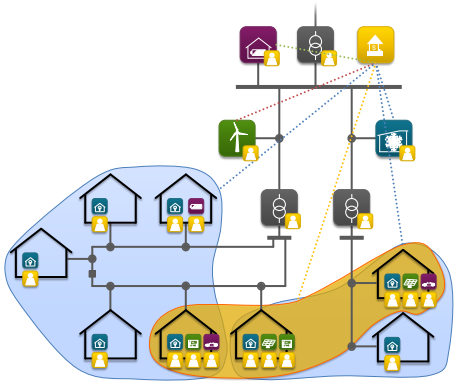

Fig. 4. Coalition Formation in the Smart Nord research project.

systems, self-organization is a promising way for providing coordination, and in turn to fulfill system-critical tasks. The self-organizing heuristic COHDA for solving combinatorial optimization problems is a building block in this vision.

References

1. Talbi, E.G.: Metaheuristics. John Wiley & Sons Inc., Hoboken, NJ, USA (2009)
2. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization. ACM Computing Surveys 35, 268–308 (2003)
3. Hinrichs, C., Lehnhoff, S., Sonnenschein, M.: A Decentralized Heuristic for Multiple-Choice Combinatorial Optimization Problems. In: Operations Research Proceedings 2012. Springer (2012), `http://www.springer.com/series/722` (to appear)
4. Hinrichs, C., Sonnenschein, M., Lehnhoff, S.: Evaluation of a Self-Organizing Heuristic for Interdependent Distributed Search Spaces. In: Filipe, J., Fred, A.L.N. (eds) ICAART 2013 - Proceedings of the 5th International Conference on Agents and Artificial Intelligence, Volume 1 - Agents, Barcelona, Spain, 15-18 February, 2013. SciTePress (2013), (accepted)
5. Jones, T.: A Description of Holland's Royal Road Function. Evolutionary Computation 2, 409–415 (1994)
6. Yang, S., Kingdom, U., Section, F.: Non-stationary problem optimization using the primal-dual genetic algorithm. In: The 2003 Congress on Evolutionary Computation, 2003. CEC '03. vol. 3, pp. 2246–2253. IEEE Press, New York (2003)
7. Rechenberg, I.: Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Fommann-Holzboog, Stuttgart (1973)
8. Smart Nord (2012), `http://www.smartnord.de`