Papachristos, S.; Ganas, I. (eds.): Proceedings of the Third ISIR Summer School on Inventory Modelling in Production and Supply Chains (Ioannina, Greece, August 1997), ISIR, 1997, pp. 155-165

MASCOT: A Multiagent System for Hybrid Optimization

Sven Hader

Department of Computer Science
Technical University of Chemnitz
D-09107 Chemnitz, Germany
E-Mail: sha@informatik.tu-chemnitz.de

Abstract

In this paper, the so-called *meta optimization problem*, that is, the task of choosing a suitable optimization technique as well as favorable strategy parameters for an optimization problem at hand, is described. It is stated that under conditions most often found in practical optimizations *hybrid* techniques are best suited. The optimization system MASCOT is introduced that uses a hybrid technique adapting dynamically to concrete problems. For illustration, the optimization of a multi-location inventory model using MASCOT is shown.

key words: hybrid optimization, multiagent systems, multi-location inventory model

1 Introduction

There has been a growing trend towards increasingly complex technical systems (e.g. logistic, manufacturing, and computer systems) in industry within the last years. Therefore, problems arise in design and modification of such systems by engineers that often cannot be solved just by domain knowledge, experience, and intuition. This results in an increasing importance of automated optimization tools that can be noticed in practice.

Most optimization problems that have to be solved in industry show a number of special characteristics (compared with „pure“ mathematical problems):

- objective function and/or constraints cannot be formulated by mathematical means but only in form of computer programs (e.g. FEM, simulation)
  → no analytical gradient information is available
  evaluation of scenarios[1] may take a long time (minutes, hours, etc.)
- large number of design variables and of linear/nonlinear constraints
  → the solution space is very large and less is known about its shape
- design variables permitted to take only discrete values with no pre-defined order relation between the values (e.g. `material` ∈ { `wood`, `copper`, `steel` })
  → gradient information has no meaning
- the solution has to be found in <u>restricted time</u>
  → only a (very) small portion of the solution space can be examined

[1] a *scenario* describes a completely specified system alternative (that is, values are assigned to all design variables)

Since there is no single optimization technique suitable for all kind of problems, the engineer has to solve a *meta optimization problem* of the form

> *„find an optimization technique (as well as favorable strategy parameters) whose application to the optimization problem provides the best solution, taking into account the restricted time (and resources)"*

before he can solve the original optimization problem. Due to the variety of existing optimization techniques and the restricted knowledge about their applicability, most engineers are incapable of solving this meta problem properly. A number of software tools exist that guide the user in choosing an appropriate technique (e.g. KBOPT [Pinto 1989]), but they do not consider the problem of restricted time.

Since any single optimization technique has its strengths and weaknesses, a combination of techniques into one *hybrid* technique seems to be advantageous. Hart says (cf. [Hart 1996]):

> *„Through hybridization, the optimization strategy can be tailored to suit the special characteristics of a problem, thereby enhancing the overall robustness and efficiency of the optimization process."*

and Sosic states (cf. [Sosic 1994]):

> *„Methods with strengths at different time intervals can be combined into hybrid methods which provide solutions in a shorter time period without sacrificing the solution quality."*

Hybrid techniques combine a number of single (also called *atomic*) optimization techniques, like steepest descent method, random search, knowledge-based optimization, genetic algorithms, etc., within one optimization run. Thereby, they often reach a higher efficiency and robustness compared to the single techniques. However, these advantages arise in many cases only from adaptation of the technique to the optimization problem (or the problem domain) considered. To adapt a hybrid technique, two main questions have to be answered:

- Which atomic techniques should be used and what are favorable strategy parameters (step sizes, termination conditions, etc.) ?
- How should the techniques interact (order of application, data exchanged, switch-over conditions, etc.) ?

Currently, there is a number of systems allowing the application of hybrid techniques to optimization problems of practical interest (e.g. EnGENEous [Powell 1991], REMO [Syrjakow 1995]). Mostly, the <u>engineer</u> has to answer the questions previously mentioned by creating some kind of *optimization plan* before the optimization run. This may overtax the engineer since he is no optimization expert usually.

In this paper, the multiagent system MASCOT is introduced. It provides a hybrid optimization technique that adapts <u>dynamically</u> to concrete problems during the optimization run. In that way, it reaches a high degree of efficiency and flexibility. MASCOT is able to solve problems from the domain of parameter optimization, especially for technical systems.

2 Multiagent Systems

Multiagent systems (MAS) are a research field of distributed artificial intelligence (DAI). They consist of collections of autonomous hardware or software systems, so-called *agents*[1], that are able to cooperate. The cooperation is coordinated by means of communication, that is, the agents can „speak to each other“. Agents respond to changes occurring in their environment (the „world“ outside the agent) in which they may take into account their local state, knowledge, skills, plans, and goals.

A common problem in MAS is to find the agent most suitable for a given task (*task allocation*). This is trivial if the task can be solved by only one agent, but may become complicated if several agents can solve it[2]. The *contract net protocol* (CNP), introduced by Smith [Smith 1980], is one approach to overcome this problem. It uses a mutual negotiation process involving a manager (the agent that needs the solution) and a set of potential contractors (agents that can generate the solution). Communication in CNP consists of the following four main steps:

(1) the manager sends a task announcement to all agents he thinks are eligible

(2) each of these agents asks itself the two questions
+ *„Do I have the skill to solve this task ?“*
+ *„Do I have the time, capacity, intent, etc. to solve this task at the moment ?“*
and, if answered positively, sends a bid message to the manager

(3) the manager evaluates the bids received, chooses the best one, and establishes a contract with the respective agent (that thus becomes the contractor)

(4) the contractor solves the task and sends the solution to the manager.

3 The MASCOT System

MASCOT (**M**ulti-**A**gent **S**ystem for the **C**ombination of **O**ptimization **T**echniques) is a system for parameter optimization based on a multiagent approach. It uses a hybrid optimization technique that can be adapted to concrete problems in the following two ways:

- static adaptation:
  before the optimization run the user can choose a set of optimization techniques to be incorporated into the system

- dynamic adaptation:
  during the optimization run the system chooses the optimization technique to be used next based on a negotiation process among the techniques available

[1] as with other fundamental terms, it is not possible to give a complete and universally accepted definition of the term *agent* (see [Wooldridge 1994])

[2] e.g., there may be several agents capable of solving a system of equations, each using a different technique

MASCOT consists of a changeable set of software agents (the *agent society*) that are realized as operating system processes and distributed over a computer network (see Figure 1). Each of the agents can perform a set of tasks and belongs to at least one agent group. The following essential groups exist:

- user interface agents
  deal with the communication between MASCOT and the user
- program interface agents
  deal with the execution of external programs, like simulations
- database agents
  deal with the storage and retrieval of data of global interest, like scenarios
- optimization agents
  perform single optimization techniques, like knowledge-based optimization, random search, genetic algorithms (GA)

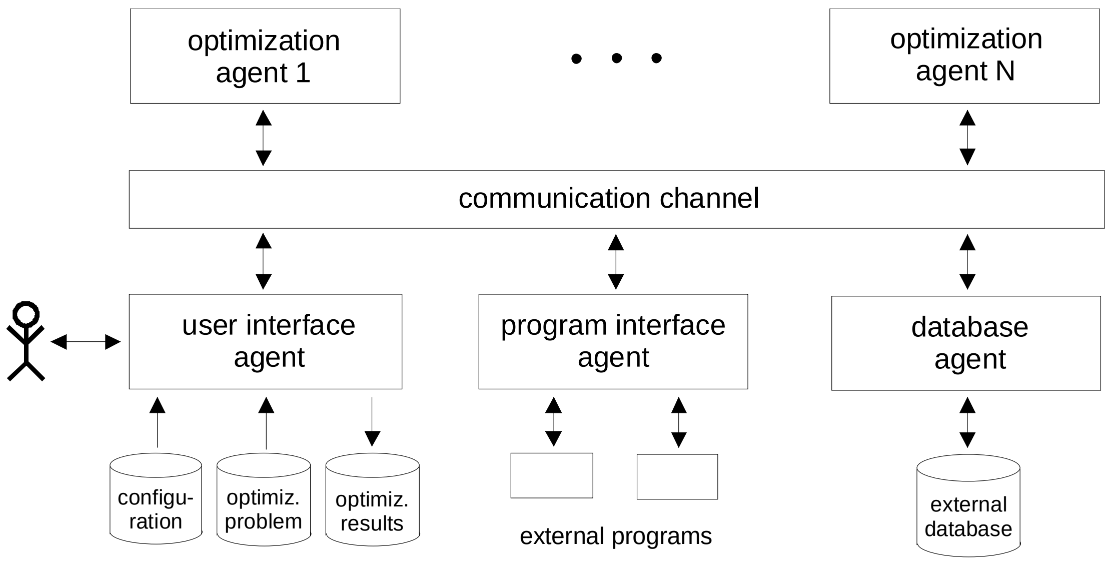

Figure 1. General architecture of MASCOT

Each of the agents consists of a communication processor, a contract processor, a task processor, and a local database (see Figure 2). The communication processor deals with the exchange of messages with other agents, providing acknowledgment and time-out mechanisms. The contract processor controls the negotiation process, that is, it evaluates announcements received, generates bids and announcements, and establishes contracts. The task processor manages all tasks executed by the agent and provides a simple task scheduler[1]. The local database contains information about the agent's skills, its current and future[2] work load, former successes/failures on tasks awarded as well as strategies for the generation of bids to „foreign" announcements and for the evaluation of bids to its own announcements.

[1] since an agent is represented by exactly one operating system process and no thread mechanism is used, tasks within a process cannot run in parallel but have to be scheduled

[2] estimation based on current work load and bids given to other agents

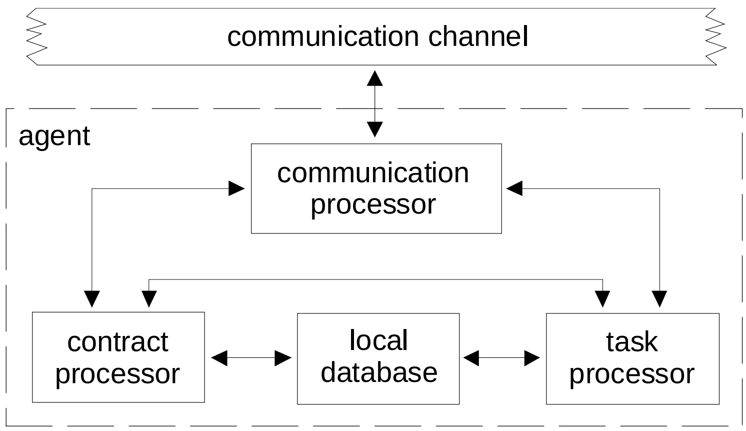

Figure 2. Structure of a MASCOT agent

Before starting MASCOT, the user has to generate a configuration file, describing the agents that should be incorporated into the multiagent system, as well as an optimization problem file, describing the problem to be solved. In addition, he has to provide a program for objective function computations (a „model program“) if the objective function cannot be expressed as a simple mathematical formula.

At start time, MASCOT is given the maximum time the optimization run is allowed to take. MASCOT reads the configuration file, starts the agents described in there, and establishes the communication channel. Then, it reads the optimization problem file, generates an internal problem description, and starts a two-step problem-solving process consisting of:

(1) finding at least one feasible scenario[1]

(2) successively improve the best scenario found so far.

The problem of finding the agent best suited for the steps (1) and (2) is solved through a negotiation process based on CNP (see Section 2). Optimization agents that received an announcement for one of the tasks evaluate whether and how well they can solve it. This is carried out by a local strategy that takes into consideration the following essential information:

- classification of the optimization problem
  (number and type of design variables, nature of objective function and constraints, etc.)
- information gathered during the optimization run
  (best scenario so far, shape of the function surface, former successes/failures, etc.)
- maximum available time for task solution
- current and future work load.

Special attention is given to the maximum available time that leads to an estimation on the number of scenarios that can be evaluated at most. Based on this number, „fast“ but rather local (e.g. hill-climbing) or „slow“ but rather global optimization techniques (e.g. GA) are preferred.

[1] this can be a serious problem in highly constrained solution spaces

Agents that can solve the task announced send their bids to the manager[1]. The manager chooses the best one, establishes a contract with the respective agent and the agent starts to solve the task. If the agent founds a solution or an assigned deadline is reached, it sends a (positive or negative) solution report to the manager and the contract is finished. Due to incomplete information at the time of negotiation, an agent may fail. In that case, the second-best, third-best, etc. agent gets its chance. If there is no alternative bidder, the task cannot be solved at all.

The task processor of an agent may decompose a task awarded and announce some or all of the resulting sub-tasks to other agents. For example, a knowledge-based optimization agent may set the values of three of the design variables, introduce two new constraints, and announce the resulting optimization problem (with a considerably smaller solution space). That is, the optimization problem may change temporarily and agents unable to solve the original problem may become applicable to a changed version of the problem.

Currently, MASCOT agents are implemented in C++ with the local databases implemented in PROLOG. The knowledge-based optimization agent is based on experiences gained in working with the DIM_EXPERTE system (cf. [Hader 1994], [Hader 1995]). The communication between agents is realized using a system called PVM (Parallel Virtual Machine, see [PVM 1994]) that allows process control and message-passing in a network of interconnected computers using TCP/IP.

4 An Example: Multi-location Inventory Model with Lateral Transshipments

Consider a single-product/multi-location inventory model with the following characteristics (see also [Köchel 1975], [Arnold 1996], [Arnold 1997]):

- $N \geq 2$ locations; infinite horizon, divided into regular periods $t = 1, 2, ...$
- at the beginning of each period t a joint order for additional inventory can be placed based on an inventory policy IP
  the ordered inventory is received immediately incurring constant ordering costs K
- during period t demand occurs at the N locations according to some random distribution that is satisfied immediately or backlogged
- at the end of period t the present local stocks can be redistributed to reduce overall costs, incurring transportation costs c_{ij} per item transported from location i to location j
- after this, holding costs h_i per item of positive stock resp. shortage costs p_i per item of „negative“ stock (backorders) are incurred for all locations $i = 1,..,N$

The problem is to find an inventory policy IP that minimizes the expected average cost of the overall system over an infinite horizon (*average optimal policy*). Köchel suggests to use a policy of the $(\sigma,\mathbf{S})$ type (cf. [Arnold 1997]), i.e., the vector of local stocks $\mathbf{x}$ is restocked to vector $\mathbf{S}$ whenever $\mathbf{x} \in \sigma$ (*order region*) holds at the beginning of a period.

[1] to ease the concept of negotiation, agents are programmed to be „benevolent“, that is, their internal goals correspond to the systems overall goal („solve this optimization problem“)

As an example, we consider a model with normal distributed demand, (σ,**S**)-type „triangle“ [1] inventory policy and the following parameters:

$N = 4$
$K = 1000$
$\mathbf{H} = (h_i) = (1, 2, 3, 4)$
$\mathbf{P} = (p_i) = (10, 9, 11, 8)$
$\mathbf{C} = (c_{ij}) = ((0, 5, 7, 4), (7, 0, 8, 5), (9, 8, 0, 7), (8, 7, 9, 0))$
demand: $\mu = (\mu_i) = (20, 30, 25, 15)$, $\sigma = (\sigma_i) = (3, 4, 2, 1)$

MASCOT was used to find optimal values for the policy parameters **s** [2] and **S** that minimize the overall cost function g(.). The following optimization techniques were combined to solve this problem cooperative: pure random search, creeping random search, steepest descent method with gradient approximation, and knowledge-based optimization. Starting with

$$\mathbf{s}_0 = (0, 0, 0, 0) \wedge \mathbf{S}_0 = (20, 30, 25, 15) \quad \rightarrow \quad g(.) = 1032.164$$

MASCOT found

$$\mathbf{s}^* = (-151, -321, -234, -254) \wedge \mathbf{S}^* = (119, 114, 71, 29) \quad \rightarrow \quad g(.) = 557.656$$

after evaluation of 1000 scenarios. Since the real optimum is not known at the moment, sound statements about the performance and efficiency of MASCOT cannot be made. Nevertheless, it has been shown that MASCOT is able to combine a set of different optimization techniques in the solution of optimization problems.

5 Conclusions

In this paper, the problem of choosing a suitable optimization technique as well as favorable strategy parameters for concrete problems that is of growing interest in industry, was described. It was stated that often a hybrid optimization technique, that is, the combination of several atomic optimization techniques within one run, is to be preferred. The optimization system MASCOT was introduced that uses a multiagent approach to realize a hybrid optimization technique for parameter optimization of technical systems. The hybrid technique used may be customized by the user before the optimization run and is adapted dynamically by the system (using a negotiation process among optimization agents) during the optimization run. As an example, a single-product/multi-location inventory model was optimized.

To improve the efficiency and robustness of the hybrid technique used, the following fields of research should be subject to further investigation: classification of optimization problems, selection of suitable optimization techniques based on this classification, selection of favorable strategy parameters for the techniques uses, improvement of the dynamic adaptation to make the most use of the restricted time available.

---

[1] for an explanation of this inventory policy see [Arnold 1997]

[2] the order region σ is described by a parameter vector $\mathbf{s} = (s_1,..,s_N)$

References

[Arnold 1996][1] Arnold, J.; Köchel, P.: *Evolutionary Optimization of a Multi-location Inventory Model with Lateral Transshipments*. Proceedings of the 9th International Working Seminar on Production Economics, Igls, Austria, 1996.

[Arnold 1997][1] Arnold, J.; Köchel, P.; Uhlig, H.: *Simulation und Evolutionäre Optimierung eines Mehrlagersystems mit Transport*. (Simulation and Evolutionary Optimization of a Multi-location Inventory System with Transshipments) Proceedings of the 6th Symposium „Simulation als betriebliche Entscheidungshilfe“, Braunlage, Germany, 1997, pp. 135-148.

[Hader 1994][2] Hader, S.: *Regelbasierte Optimierung von Produktionssystemen*. (Rule-based Optimization of Manufacturing Systems), Proceedings of the 9th Symposium „Simulationstechnik“, Stuttgart, Germany, 1994, pp. 603-606.

[Hader 1995][2] Hader, S.: *A Knowledge-based Approach to the Optimization of Technical Systems and Its Application to Inventory Systems*. Proceedings of the International Postgraduate Summer School on Inventory Research, Portoroz, Slovenia, 1995, Vol. 2, pp. 87-92.

[Hart 1996][3] Hart, W.E.; Eldred, M.S.: *Hybrid global/local algorithm research*. 1996.

[Köchel 1975] Köchel, P.: *Ein stochastisches Lagerhaltungsmodell für mehrere miteinander verbundene Lager*. (A Stochastic Inventory Model with Several Connected Locations), Math. Operationsforschung und Statistik, 6(1975)3, pp. 413-426.

[Pinto 1989] Pinto, I.C.: *Wissensbasierte Unterstützung bei der Lösung von Optimierungsaufgaben*. (Knowledge-based Support to the Solution of Optimization Problems), PhD thesis, University of Dortmund, Department of Computer Science, 1989.

[Powell 1991] Powell, D.J.; Skolnick, M.M.; Tong, S.S.: *Interdigitation: A Hybrid Technique for Engineering Design Optimization Employing Genetic Algorithms, Expert Systems, and Numerical Optimization*. In: Davis, L. (ed.): Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York, 1991, pp. 312-331.

[PVM 1994][4] *PVM 3 User's Guide and Reference Manual*. Oak Ridge National Laboratory, Oak Ridge, Tennessee, USA, 1994.

[Smith 1980] Smith, R.G.: *The Contract Net Protocol: High Level Communication and Control in a Distributed Problem Solver*. IEEE transactions on computers, 29(1980)12, pp. 1104-1113.

[Sosic 1994][5] Sosic, R.; Wilby, G.D.: *Using the Quality-Time Tradeoff in Local Optimization*. Proceedings of the IEEE Second ANZIIS Conference, Brisbane, Australia, 1994, pp. 253-257.

[Syrjakow 1995] Syrjakow, M.; Szczerbicka, H.: *Simulation and Optimization of Complex Technical Systems*. Proceedings of the SCSC '95, Ottawa, Ontario, Canada, 1995, pp. 86-95.

[Wooldridge 1994] Wooldridge, M.; Jennings, N.R.: *Agent Theories, Architectures, and Languages: A Survey*. Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Amsterdam, The Netherlands, 1994, pp. 1-39.

---

[1] available via `http://www.tu-chemnitz.de/~jarn`

[2] available via `http://www.tu-chemnitz.de/~hader`

[3] available via `http://endo.sandia.gov/9234/sd_optim/f2_hybrid.html`

[4] available via `http://www.epm.ornl.gov/pvm/pvm_home.html`

[5] available via `http://www.cit.gu.edu.au/~sosic/nqueens.html`