ARTÍCULO

Optimal scheduling problem for computing agent systems

Maciej Smołka*

Institute of Computer Science
Jagiellonian University
Kraków, Poland
smolka@ii.uj.edu.pl

Abstract

The paper presents a mathematical model of a computing multi-agent system. The dynamics of a MAS is described in terms of stochastic process theory. Then the optimal scheduling problem for the MAS is formulated together with a result on the existence of optimal strategies. Finally some sufficient conditions for optimality are presented as a way of verification of heuristic scheduling strategies.

Keywords: Distributed computing, mobile agents, task scheduling.

1 Motivation

Multi-agent paradigms are slowly starting to prove their efficiency in design and implementing of large distributed computational systems, although it is still not the mainstream of MAS applications. It appears that MAS paradigms provide a fertile ground for theoretical considerations on the problem of task scheduling in distributed computations. The idea of splitting and migrating tasks tells us to look for analogies with e.g. fluid dynamics. The first step is to formulate some heuristic scheduling strategies (see [5, 2]) based on the idea of task diffusion. The diffusion-based scheduling strategy has proven to be simple and efficient (see [2, 7]) in implementing various computational problems ([8, 1, 9, 6]) but it lacks a theoretical background allowing us to determine if it is optimal or quasi-optimal in any sense. Thus there was a need to build a formal model for multi-agent computations to provide a way for verification of empirically invited strategies. Such a model was first presented in [7] and in this paper we present its developed version together with some theoretical results concerning existence of optimal strategies and sufficient conditions for optimality. The model is presented in terms of stochastic control theory.

The description of the dynamics of a MAS is contained in section 2. First we introduce a special quantity describing the state of our system as a whole. Then we formulate equations of evolution for that quantity, which allow us to treat the MAS as a stochastic process. Then we show that in many natural situations it can be even a Markov chain. In section 3 we formulate an optimal control problem for our MAS together with a result on the existence of optimal solutions. We give some important examples of cost functionals. Finally we present Bellman-type optimality

*The author has been supported by the State Committee for Scientific Research of the Republic of Poland under research grants 2 P03A 003 25 and 7 T07A 027 26

conditions, which may be used for verification of any heuristic scheduling strategy.

2 System dynamics

2.1 Notation and preliminaries

In this section we state a formal mathematic model for multi-agent computations. It is based on the architectural principles of computing multi-agent system as defined on the Octopus platform (described in [8, 3]) as well as the experience in implementing computational agent applications on the platform and, especially, a heuristic diffusion-based scheduling strategy for such applications.

Our model was presented first in [7] and we shall use the notations introduced therein. Here we present the main concepts of the model with some minor changes. Consider that we are given a set of all possible agents and denote it by $\mathbf{A}$. Of course in reality the determination of such a set can be rather problematic. It is one of the reasons to avoid considering the evolution of a single agent. Instead, we shall study the dynamics of a whole computing multi-agent system. In this paper we shall consider its discrete-time evolution. Recall the notion of the *vector weight of an agent* (cf. [7]) which is the mapping

$$w : \mathbb{N} \times \mathbf{A} \longrightarrow \mathbb{R}^2_+$$

whose components are the agent task remaining time measured in common units and the agent memory requirement in bytes (the paper [7] uses also symbols E_i and M_i). Assume that the dependency of the total weight of child agents after partition upon their parent's weight before partition is well-known, componentwise and linear, i.e. we know the constants $c_1, c_2 \geq 0$ such that in the case of partition $A \to \{A_1, A_2\}$ we have

$$w^i_{t+1}(A_1) + w^i_{t+1}(A_2) = c_i w^i_t(A)$$

for $i = 1, 2$ (note that in comparison to [7] we assume that the change of weight during partitions is *linear*). Such an assumption seems realistic, in simple cases we have $c_i = 1$ but in general the constants may be either greater or less than 1.

Next recall the notion of the *total weight of all agents allocated on a virtual node P at any time* t, i.e.

$$W_t(P) = \sum_{Sch_t(A)=P} w_t(A)$$

(notations borrowed from [2]). Obviously we put 0 if no agent is maintained by P. This notion is crucial in our search for a global description of the system dynamics since it is a global quantity describing the state of the system in such a way that seems appropriate for our purposes.

In the sequel we shall assume that the number of virtual nodes

$$\sharp\mathbf{N} = N$$

is *fixed*. Thus we can consider W_t as a nonnegative vector in $\mathbb{R}^{2N}$ such that

$$\begin{cases} W^j_t & = W^1_t(P_j) \\ W^{N+j}_t & = W^2_t(P_j) \end{cases}$$

for $j = 1, \ldots, N$.

2.2 State equations

As said before we shall consider W_t as a state of a computing application. Now we shall formulate the equations of evolution of W_t. Because of the nondeterministic nature of some computations (see e.g. [9, 6]) and the need to involve the influence of an operating system and networking environment it is necessary to include a stochastic perturbation in our state equations. In other words, we shall treat W_t as a *stochastic process on* $\mathbb{R}^{2N}_+$.

First consider three simple cases.

'Established' evolution (without migrations or partitions). Then the state equation has the form

$$W_{t+1} = F_t(W_t)$$

where F_t is a given stochastic nonnegative vector field.

Partition at node j. Then we have

$$\begin{cases} W^j_{t+1} & = \big(1 - u^1_{jj}(W_t)\big)\, W^j_t \\ & \quad + c_1\ u^1_{jj}(W_t)\ W^j_t \\ W^{N+j}_{t+1} & = \big(1 - u^2_{jj}(W_t)\big)\, W^{N+j}_t \\ & \quad + c_2\ u^2_{jj}(W_t)\ W^{N+j}_t \\ W^i_{t+1} & = W^i_t \qquad \text{for } i \notin \{j, N+j\} \end{cases}$$

where $u^m_{jj} : \mathbb{R}^N_+ \to [0, 1], m = 1, 2$ are the proportions of the weight components of splitting agents to the corresponding components of the total weight of all agents at node j.

Migration from j to k. In this case the state equations have the form

$$\begin{cases} W^j_{t+1} &= \Big(1 - u^1_{jk}(W_t)\Big)\, W^j_t \\ W^{N+j}_{t+1} &= \Big(1 - u^2_{jk}(W_t)\Big)\, W^{N+j}_t \\ W^k_{t+1} &= W^k_t + u^1_{jk}(W_t) W^j_t \\ W^{N+k}_{t+1} &= W^{N+k}_t + u^2_{jk}(W_t) W^{N+j}_t \\ W^i_{t+1} &= W^i_t \qquad \text{for } i \notin \{j, k, N+j, N+k\} \end{cases}$$

with u^m_{jk} analogous to u^m_{jj}.

Of course in reality we usually have a mix of the three cases. Putting them all together we obtain the following state equations

$$\begin{cases} W^i_{t+1} &= F^i_t(\tilde{W}_t) + c_1\ u^1_{ii,t}(W_t)\ W^i_t \\ & \quad + \sum_{j \neq i} u^1_{ji,t}(W_t)\ W^j_t \\ W^{N+i}_{t+1} &= F^{N+i}_t(\tilde{W}_t) + c_2\ u^2_{ii,t}(W_t)\ W^{N+i}_t \\ & \quad + \sum_{j \neq i} u^2_{ji,t}(W_t)\ W^{N+j}_t \end{cases} \tag{1}$$

for $i = 1, \ldots, N$, where

$$\begin{cases} \tilde{W}^i_t &= \Big(1 - \sum_{k=1}^N u^1_{ik,t}(W_t)\Big)\, W^i_t \\ \tilde{W}^{N+i}_t &= \Big(1 - \sum_{k=1}^N u^2_{ik,t}(W_t)\Big)\, W^{N+i}_t \end{cases}$$

with initial conditions

$$W_0 = \hat{W}. \tag{2}$$

Note that in equation (1) we have also added the explicit time dependency of u^m_{ij}. It follows that our W_t is a *controlled stochastic process* with a *control strategy*

$$\pi = (u^1_t, u^2_t)_{t \in \mathbb{N}} \tag{3}$$

such that

$$u^m_t : \mathbb{R}^{2N}_+ \longrightarrow U. \tag{4}$$

The control set U contains matrices from $[0,1]^{N \times N}$ that satisfy at least the following conditions

$$\begin{cases} u_{ij} \cdot u_{ji} = 0 & \text{for } i \neq j, \\ \sum_{k=1}^N u_{ik} \leq 1 & \text{for } i = 1, \ldots, N. \end{cases} \tag{5}$$

In fact quite often the conditions imposed on U shall be more restrictive (see next subsection).
The first equation in (5) can be interpreted in the following way: *at a given time migrations between two nodes may happen in only one direction.* The second equality says that *the number of agents leaving a node must not exceed the number of agents present at the node just before the migration.*

Remark. It is easy to see that the control set U defined by the conditions (5) is compact (and so are of course its closed subsets).

To give the state equations a more concise form let us introduce some additional notations. Denote by $\mathrm{diag}(x)$ the $n \times n$ matrix obtained by putting the components of the vector $x \in \mathbb{R}^n$ on the diagonal and 0 outside the diagonal. Denote by $\mathrm{d}(A)$ the matrix constructed from a matrix A by setting its non-diagonal terms to 0. Denote by $\mathbf{1}$ the vector from $\mathbb{R}^n$ composed of 1's. Finally denote

$$C = \left[\begin{array}{c|c} c_1 I_N & 0 \\ \hline 0 & c_2 I_N \end{array}\right] \in \mathbb{R}^{2N \times 2N}_+$$

where I_n is the $n \times n$ identity matrix and

$$u = \left[\begin{array}{c|c} u^1 & 0 \\ \hline 0 & u^2 \end{array}\right] \in \mathbb{R}^{2N \times 2N}_+.$$

With the above notation our state equations look like

$$W_{t+1} = F_t\Big(\big(I_{2N} - \mathrm{diag}(u_t(W_t) \cdot \mathbf{1})\big)\ W_t\Big) + \Big[(C - I_{2N})\, \mathrm{d}(u_t(W_t)) + u^T_t(W_t)\Big] W_t. \tag{6}$$

2.3 State space details

In the most general case one could take the whole $\mathbb{R}^{2N}_+$ for the state space of the stochastic process W_t. But in real situations it is unnecessary and impractical. First of all real computers have limited memory so we have

$$0 \leq W^{N+i}_t \leq W^{N+i}_{max} \tag{7}$$

for $i = 1, \ldots, N$. On the other hand it is quite safe to assume that we are able to estimate roughly the maximum time of computations on every machine, namely

$$0 \leq W^i_t \leq W^i_{max}. \tag{8}$$

Moreover from the considerations of the subsection 2.1 it follows that we may assume that the components of W_t are integers. Putting it all together we obtain the following state space

$$S = \left\{0, \ldots, W^1_{max}\right\} \times \cdots \times \left\{0, \ldots, W^{2N}_{max}\right\}. \tag{9}$$

Let us call the elements of this finite set s_i, i.e.

$$S = \{s_0 = 0, s_1, \ldots, s_K\}.$$

This analysis of the state space has some consequences. First of them is that we should impose the following condition on F

$$F_t : S \times \Omega \longrightarrow S \tag{10}$$

where (Ω, Σ, P) is a probability space. Likewise, the condition (5) is not sufficient for the equations (1) (or (6)) to make sense. Namely we have to assume that for any $t \in \mathbb{N}$ and $W \in S$

$$u_t(W) \in U_W$$

where U_W is the set of all $2N \times 2N$ matrices u such that

$$\begin{cases} u^m \in U \quad \text{for } m = 1, 2 \\ u^1_{ij}(W) \in \left\{ \dfrac{k}{W^i} : k = 0, \ldots, W^i \right\} \\ u^2_{ij}(W) \in \left\{ \dfrac{k}{W^{N+i}} : k = 0, \ldots, W^{N+i} \right\} \\ u^1_{ij}(W) = 0 \quad \text{if } W^i = 0 \\ u^2_{ij}(W) = 0 \quad \text{if } W^{N+i} = 0 \end{cases} \tag{11}$$

Finally assume that F_t is such that the right hand side of (6) belongs to S for any admissible control.

3 Optimal scheduling problem

3.1 Formulation

In the sequel we shall assume that the mapping F has the form

$$F_t(W) = F(W, \xi_t) \tag{12}$$

where $\xi_0, \xi_1, \ldots$ are mutually independent identically distributed random variables. Denote by $G(W, u, \xi)$ the right hand side of the equation (6), i.e.

$$G(W, u, \xi) = F\Big(\big(I_{2N} - \operatorname{diag}(u \cdot \mathbf{1})\big)\ W,\ \xi\Big) + \Big[(C - I_{2N})\,\mathrm{d}(u) + u^T\Big]W. \tag{13}$$

Remark. Given (9) and (12) it is easy to show that W_t is a *controlled Markov chain* with transition probabilities

$$p_{ij}(\alpha) = P\Big(G(s_i, \alpha, \xi_0) = s_j\Big) \tag{14}$$

for $i, j = 0, \ldots, K$, $\alpha \in U_{s_i}$. The transition matrix for the control u is

$$P(u) = \Big[p_{ij}(u(s_i))\Big]_{i,j=0,\ldots,K}. \tag{15}$$

In the following considerations we shall assume that

$$F(0, \xi) = 0 \tag{16}$$

with probability 1. This is quite natural condition saying that in the case of no migrations or partitions our computing system cannot leave 0 (its desired final state). Then for any $\alpha \in U_0$ we have

$$G(0, \alpha, \xi) = F(0, \xi) = 0$$

with probability 1, which means that the computations cannot leave 0 even if we apply some agent operations. In stochastic process terminology we say that 0 is an *absorbing state* of Markov chain W_t.

The general form of the cost functional for controlled Markov chains of our type is (cf. [4])

$$V(\pi; s) = E\left[\sum_{t=0}^{\infty} k(W_t, u_t(W_t))\right] \tag{17}$$

where π is a control strategy (3) and s is the initial state of W_t, i.e.

$$W_0 = s \tag{18}$$

Since 0 is an absorbing state we shall always assume that

$$k(0, \cdot) = 0, \tag{19}$$

i.e. remaining at 0 has no cost (it guarantees that the overall cost can be finite).

Recalling subsection 2.3 we define the following set of admissible controls

$$\mathbf{U} = \big\{\pi : u_t(W) \in U_W, t \in \mathbb{N}\big\}. \tag{20}$$

Now we are in position to formulate the *optimal scheduling problem*. Namely given an initial configuration s_i we search for such control strategy $\pi^* \in \mathbf{U}$ that

$$V(\pi^*; s) = \min\big\{V(\pi; s) :\ \pi \in \mathbf{U},\ W_t \text{ is a solution of } (6), (18)\big\}. \tag{21}$$

3.2 Examples of cost functionals

Consider some cost functionals which seem appropriate for multi-agent computations. The first one is the expected total time of computations

$$V_T(\pi; s) = E\big[\inf\{t \geq 0 : W_t = 0\} - 1\big]. \tag{22}$$

We can rewrite it in the form (17) if we put

$$k(s_j, \alpha) = 1 \quad \text{for } j \neq 0,\ \alpha \in U_{s_j}.$$

The second example takes into account the mean load balancing over time. It has the following form

$$V_L(\pi; s) = E\left[\sum_{t=0}^{\infty}\sum_{i=1}^{N}(L_t^i - \overline{L}_t)^2\right] \tag{23}$$

where

$$L_t^i = \frac{W_t^i}{perf_i(W_t^{N+i})}$$

is the load concentration (for explanations on functions *perf* and the above formula see [2]) and

$$\overline{L}_t = \frac{1}{N}\sum_{i=1}^{N} L_t^i$$

is its mean over all computing nodes. This time the form of k is straightforward.

Both the above examples do not contain an explicit dependency on the control. Generalising it a little allows us to put some cost on migrations. Namely take $a \geq 0$, $\varrho_{ij} \geq 0$ and $\mu_{ij} : [0, 1] \to \mathbb{R}_+$ nondecreasing and such that $\mu_{ij}(0) = 0$, and put

$$k_M(s_j, \alpha) = \varphi(s_j) + a\sum_{i\neq j}\varrho_{ij}\mu(\alpha_{ij})$$

and

$$V_M(\pi; s) = E\left[\sum_{t=0}^{\infty} k_M(W_t, u_t(W_t))\right]. \tag{24}$$

3.3 Existence of optimal strategies

Now let us consider the existence of solutions for problem (21). To this end let us denote by by

$$R(u) = \Big[p_{ij}(u(s_i))\Big]_{i,j=1,\dots,K}.$$

the 'probably not absorbing' part of the transition matrix for a control u and by $R^n(u)$ the analogous part of n-step transition matrix obtained by applying the stationary control $u_0 = \cdots = u_{n-1} = u$. It is easy to see that

$$R^n(u) = R(u)\dots R(u).$$

The following proposition is a straightforward consequence of [4, Theorem 4.2].

Proposition 3.1. *Assume that*

1. *$R^K(u)$ is a contraction for every u such that $u(s) \in U_s$ or*
2. *$R^n(u)$ is a contraction for some $n \geq 1$ and u as above but additionally*

$$k(s_j, \alpha) \geq \varepsilon > 0$$

for $j \neq 0$, $\alpha \in U_{s_j}$.

Then there exists the unique optimal solution of (21).

Proof. It is sufficient to notice that U_s are finite so assumptions (A1)–(A3) from [4, Chapter 4] are satisfied in a straightforward manner. □

Corollary 3.2. *Consider our example cost functionals from the previous subsection.*

1. *Problem* (21) *for V_T has the unique solution provided the assumption 2 holds.*
2. *Problem* (21) *for V_L has the unique solution provided the assumption 1 holds.*
3. *Existence of the solution for* (21) *for V_M depends on the assumption on φ. If the latter is separated from* 0 *we need assumption 2 otherwise assumption 1.*

3.4 Optimality conditions

Now we shall present some Bellman-type optimality conditions for (21). We shall formulate them in the following proposition.

Proposition 3.3. *Make the same assumptions as in Proposition 3.1. Then the optimal solution of* (21) *is a stationary strategy*

$$\pi^* = u^\infty = (u, u, \dots)$$

and it is the unique solution of the equation

$$V(\pi^*; s) = \min_{\alpha\in U_s}\left[\sum_{j=1}^{K} p_{ij}(\alpha)V(\pi^*; s_j) + k(s, \alpha)\right]. \tag{25}$$

The solution of (25) *exists and is the optimal solution of* (21).

Proof. It is another consequence of [4, Theorem 4.2] and its proof. □

To put (25) in a vector form denote

$$V = \begin{bmatrix} V(\pi^*; s_1) \\ \vdots \\ V(\pi^*; s_K) \end{bmatrix}, K(u) = \begin{bmatrix} k(s_1, u(s_1)) \\ \vdots \\ k(s_K, u(s_K)) \end{bmatrix}.$$

Then we have

$$V = \min_{w: w(s) \in U_s} \big[R(w)V + K(w)\big] = R(u)V + K(u). \tag{26}$$

Remark. Equations (25) or (26) provide us with a first tool to check our heuristic strategies for optimality. They allow us also to construct the optimal strategy by means of some iterative procedures (like Gauss-Seidel). But this of course may appear to be too expensive to consider in practice.

4 Conclusions and further research

The presented mathematical model is an attempt to provide a solid basis for the problem of scheduling computational tasks in a distributed environment. We have formulated the problem in terms of stochastic optimal control theory. Some results on the existence of optimal solutions as well as sufficient conditions for optimality have been obtained as a consequence of using this machinery. We can use this conditions to estimate how far our heuristic scheduling strategies are from the optimum. Much more are expected, first of all we search for a local form of optimality conditions. There is also a set of related practical problems to consider, e.g. one needs to compute the quantities appearing in equations (1).

References

[1] Marek Grochowski, Robert Schaefer, and Piotr Uhruski. An agent-based approach to a hard computing system — Smart Solid. In *Proceedings of the International Conference on Parallel Computing in Electrical Engineering (PARELEC 2002)*, pages 253–258, Warsaw, Poland, 22–25 September 2002. IEEE Computer Society Press.

[2] Marek Grochowski, Robert Schaefer, and Piotr Uhruski. Diffusion based scheduling in the agent-oriented computing systems. *Lecture Notes in Computer Science*, 3019:97–104, 2004.

[3] Marek Grochowski, Robert Schaefer, and Piotr Uhruski. Octopus — computation agents environment. *Inteligencia Artificial*, 2005. submitted.

[4] Harold Kushner. *Introduction to Stochastic Control*. Holt, Rinehart and Winston, 1971.

[5] Emilio Luque, Ana Ripoll, Ana Cortés, and Tomàs Margalef. A distributed diffusion method for dynamic load balancing on parallel computers. In *Proceedings of EUROMICRO Workshop on Parallel and Distributed Processing*, pages 43–50, San Remo, Italy, January 1995. IEEE Computer Society Press.

[6] Jolanta Momot, Krzysztof Kosacki, Marek Grochowski, Piotr Uhruski, and Robert Schaefer. Multi-agent system for irregular parallel genetic computations. *Lecture Notes in Computer Science*, 3038:623–630, 2004.

[7] Maciej Smołka, Marek Grochowski, Piotr Uhruski, and Robert Schaefer. The dynamics of computing agent systems. *Lecture Notes in Computer Science*, 3516:727–734, 2005.

[8] Piotr Uhruski, Marek Grochowski, and Robert Schaefer. Multi-agent computing system in a heterogeneous network. In *Proceedings of the International Conference on Parallel Computing in Electrical Engineering (PARELEC 2002)*, pages 233–238, Warsaw, Poland, 22–25 September 2002. IEEE Computer Society Press.

[9] B. Wierzba, A. Semczuk, Joanna Kołodziej, and Robert Schaefer. Hierarchical genetic strategy with real number encoding. In *Proceedings of 6th Conference on Evolutionary Algorithms and Global Optimization*, pages 231–237, Łagów Lubuski, Poland, 2003. Warsaw University of Technology Press.