Properties of the Task Allocation Problem*

J.F. de Ronde, A. Schoneveld and P.M.A. Sloot

September 6, 1996

Parallel Scientific Computing and Simulation Group
Faculty of Mathematics, Computer Science, Physics & Astronomy
University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam
The Netherlands
Phone: +31 20 5257463
fax: +31 20 5257490
E-mail: {janr, arjen, peterslo}@wins.uva.nl
http://www.fwi.uva.nl/research/pscs/

*The frontpage shows cross-sections of the landscape of the task allocation problem. Its shape is clearly influenced by variation of γ.

Contents

1 Introduction

An essential problem in the field of parallel computing is the so called *Task Allocation Problem*(TAP): given a set of parallel communicating tasks (a parallel application) and a parallel distributed memory machine, find the optimal allocation of tasks onto the parallel system. The quality of an allocation is measured by the turn-around time of the application, which depends on various components. In a parallel application, generally, one can distinguish phases dominated by communication components and calculation components.
A method that is used to minimise the turn-around time must optimise both components simultaneously. This is due to the fact that the two terms can not be regarded as independent components, rather they are strongly related to each other. A task allocation where all tasks are placed on a single processor obviously minimizes the amount of communication, while the calculation will be maximal. On the other hand equal distribution of the set of parallel tasks, without taking into account the communication term would lead to an optimised calculation term, while communication can become degraded. We toss the term *frustration* for the fact that optimisation of one term conflicts with the other, in analogy to physical systems that exhibit frustration. Intuitively, it is clear that increasing dominance of either the communication or calculation term reduces the amount of frustration in the system.

Many fundamental problems from natural sciences deal with *complex systems*. A complex system can be described as a population of unique elements with well defined attributes and interactions. In most cases, such systems are characterised by *quenched* disorder and *frustrated*, non-linear interactions [13], between the set of elements constituting the system. It is well known that these system ingredients contribute to the emergent unpredictable behaviour that is often demonstrated by such systems [16]. The quenched disorder is either present in the initial condition (e.g. in cellular automata) or in the interaction between elements (e.g. in spin glasses). In combination with the frustration occurring due to mutual conflicts, certain properties of these systems are often analytically intractable. Examples of such properties are its asymptotic behaviour and the exact location of the (energetically) optimal states. The latter characteristic often causes the corresponding optimisation problems to be NP-hard [17]. Given the fact that the TAP objective function (minimization of turn-around time) contains two competitive terms, behaviour similar to other known *complex systems* is to be expected.
In order to deepen our knowledge about the TAP, we intend to explore its characteristics in terms of phase space and optima structure. Specifically, the degree of frustration in the TAP constitutes a fundamental difficulty with the problem. An important distinguishing aspect in the TAP is the presence of a transition from sequential to parallel optimal allocation. For example, consider a parallel machine topology consisting of identical processors, with a tuneable performance rate. Increasing the peak performance continuously from 0 flop/s to ∞ flop/s, will induce a transition from optimal parallel- to sequential allocation, given a finite communication speed within the network. In analogy with other combinatorial optimisation problems that exhibit frustration and phase transitions, we expect that a phenomenon, known as *critical slowing down*, can be observed in the transition region. That is, the difficulty of finding optimal solutions peaks near the transition region (see e.g. [22]).

In general, the selection of a suited heuristic method for finding (sub)-optimal solutions requires knowledge of the shape of the phase space. Great care has to be taken in selecting an optimisation method, since searching the optimal solution to the TAP is known to be an NP-hard problem [1]. Hence, a study on the structure of the landscape of the TAP is necessary in order to identify effective optimisation methods. Furthermore, the sensitivity of the TAP to a small set of machine and application specific parameters is investigated. We restrict our attention to a specific subset of the TAP. The focus will be on applications that can be described by static parallel task graphs. In addition we assume to have a static resource parallel machine that is homogeneous and fully connected.

This paper is structured as follows. Section 2 introduces application and machine representations

that are used to model the performance characteristics of parallel static applications on parallel machines. Section 3 gives a detailed study on the structure of the phase space (or landscape) of the TAP. Section 4 is dedicated to the geometrical phase transition occurring in the TAP. In section 5 the following experimental methods are presented: Simulated Annealing (SA) [8], for finding optima, and Weinberger correlation for phase space structure characterisation [21]. In section 6 experimental results are presented, which are discussed in section 7. Finally, some concluding remarks and directions for future work are given in section 8.

2 Application and Machine Models

In order to facilitate our study on abstract parallel applications we introduce a random graph representation as a model of static communicating parallel tasks. Each task is assigned a workload and every pair of tasks (vertices) in the task graph is connected with a probability γ ($\gamma \in [0,1]$). A message size is assigned to each link between two communicating tasks. We restrict our attention to constant work loads and message sizes. Furthermore the target processor topology is assumed to be a static parallel machine that is fully connected and homogeneous. That is, communication channels between all processor pairs are bi-directional and have equal bandwidths. Moreover, the processors are homogeneous, i.e. they have identical constant performance.
The metric for deciding on the quality of a task allocation is the turn-around or execution time. A variety of cost models that are based on a graph representation can be found in literature. For example, the following cost function (1) [7], is known to model the actual execution time for a given task allocation with reasonable accuracy. Of course it is a simplification of the real situation, i.e. message latencies and network congestion are neglected.

$$H = \max_{q \in \mathcal{Q}} \left(\sum_{u_i \in \mathcal{U}^q} W_{u_i} S_q + \max_{u_i \in \mathcal{U}^q, u_j \in \mathcal{A}(u_i)} S_{pq} W_{u_i u_j} \right), \tag{1}$$

where

- u_i is a task in the parallel task graph
- $\mathcal{Q}$: the set of processors
- $\mathcal{A}(u_i)$: the set of tasks connected to task u_i
- $\mathcal{U}^q$: set of tasks u_i residing on processor q
- W_{u_i}: Work associated with task u_i (e.g. in terms of flop)
- S_q: $\frac{1}{processorspeed}$ for processor q (e.g. in s/flop)
- $W_{u_i u_j}$: Number of bytes to be sent, due to nodal connectivity, between host processor of task u_i and task u_j.
- S_{pq}: $\frac{1}{bandwidth}$ of route between processor p and q (in s/bytes)

A property of this specific function is that the execution time is determined by the "slowest" processor in the parallel machine. This cost function is a reasonable representation of the actual execution time.
Because the value of H (Eq. 1) can only change in task transfers that involve the slowest processor, it is not very sensitive to task rearrangements. Therefore it is unsuitable for local search optimisation techniques like SA. Usage of SA for finding optimal solutions necessitates formulation of an alternative cost function like (2), see e.g. [11].

$$H = \sum_p W_p^2 + \mu \sum_{p \neq q} C_{pq} \quad , \tag{2}$$

where

- $W_p = A_p S_p$, with A_p: $\sum_{u_i \in \mathcal{U}^p} W_{u_i}$, total work on processor p in terms of flop.
- $C_{pq} = M_{pq} S_{pq}$, with M_{pq}: $\sum_{u_i \in p, u_j \in q} W_{u_i u_j}$.
- μ is a control parameter, expressing the communication/calculation ratio [4].

An incremental search from a given allocation (moving one task), requires a complete re-calculation of the cost for Eq. 1. On the other hand, Eq. 2 has the locality property, which means that incremental changes in a task allocation can be propagated into the cost without having to recalculate the whole cost function. Only a difference has to be calculated instead[10]. This is specifically useful if an optimisation algorithm is applied that is based on incremental changes (e.g. SA), and as such can exploit the direct consequence of these increments. A disadvantage of using (2) is the fact that it is a not a correct model for the absolute cost. The objective is to minimise the variance in the workload distribution simultaneous with the communication volume of the whole system, opposed to optimisation of the execution time of the slowest processor in Eq. 1.

3 Correlation Structure of the TAP

The configuration space C of the TAP consists of all possible task allocations of the n tasks to the P processor topology. A configuration can be encoded as a sequence of length n, which is composed of letters taken from the alphabet $\{1, 2, ..., P\}$. The index of a sequence letter corresponds to a task ID. The distance is given by the number of positions in which two sequences A and B differ; this metric distance measure is the *Hamming distance* [5] $d(A, B)$. The *Hamming graph* can be constructed by connecting every sequence pair (A, B) that has $d(A, B) = 1$.
The number of configurations with a given distance d from an arbitrary reference point $N(P, n, d)$, the total number of configurations $\#C$, and the diameter in the configuration space, $diamC$ are easily found to be:

$$N(P, n, d) = \left(\begin{array}{c} n \\ d \end{array} \right) (P-1)^d \tag{3}$$

$$\#C = P^n \tag{4}$$

$$diamC = n \tag{5}$$

A random walk through some landscape, can be used to characterise its structure [21]. For landscapes that are self-similar it is known that the corresponding random walk auto-correlation function is a decaying exponential, with correlation length λ. Such landscapes are classified as AR(1) landscapes and have been identified in various fields, e.g. (Bio)physics [21] and combinatorial optimisation [19][20]. It has been shown that incremental search methods like Simulated Annealing perform optimally on landscapes that show a self-similar structure [18].
We will derive expressions for the relaxation and auto-correlation functions of random walks through the task allocation landscape. The relaxation functions indicate at what rate a random walk through the Hamming graph deviates from the starting point, analogous to e.g. relaxation of diffusion processes in physical systems.
The auto-correlation function is used to quantify the rugged-ness [21] of the landscape of the TAP. The landscape constitutes the Hamming graph with cost values that are assigned to all vertices according to Eq. 2. Using these expressions it is shown that the landscape is AR(1) with a correlation length that is linearly proportional to the number of tasks n.

3.1 Relaxation of random walks

The statistical properties of random walks on the graph C are completely contained in the probabilities ϕ_{sd}, where ϕ_{sd} denotes the probability that a random walk is within a distance d from the

starting point after s steps. In general this probability distribution fulfills the following recursion relations on any distance transitive graph (following [19]).

$$\begin{aligned} \phi_{sd} &= a^+_{d-1}\phi_{s-1d-1} + a^0_d\phi_{s-1d} + a^-_{d+1}\phi_{s-1d+1} \\ \phi_{00} &= 1 \\ \phi_{sd} &= 0, \qquad \text{if} \qquad d > s \end{aligned} \tag{6}$$

The coefficients a^+_d, a^0_d and a^-_d denote the probability of making a step "forward", "sideward" and "backward", respectively, given the walk is within a distance d from the starting point. Therefore $a^+_d + a^0_d + a^-_d$ is equal to 1. For the TAP graph C we obtain the following expressions:

$$\begin{aligned} a^+_d &= \frac{(n-d)(P-1)}{nP} \\ a^0_d &= \frac{n+(P-2)d}{nP} \\ a^-_d &= d/(nP) \end{aligned} \tag{7}$$

Although we have no closed expression for the ϕ_{sd}, we can obtain some insight into the relaxation behaviour of random walks from the expected values of the distance (first moment) and the squared distance (second moment) from the starting point after s steps along the walk:

$$\begin{aligned} \Delta_1(s) &= \sum_{d=0}^{s} d\phi_{sd} \\ \Delta_2(s) &= \sum_{d=0}^{s} d^2\phi_{sd} \end{aligned} \tag{8}$$

Using Eqs. 6 and 8, we can derive recursion relations for $\Delta_1(s)$ and $\Delta_2(s)$:

$$\begin{aligned} \Delta_1(s) &= \sum_{d=0}^{s} d(a^+_{d-1}\phi_{s-1,d-1} + a^0_d\phi_{s-1,d} + a^-_{d+1}\phi_{s-1,d+1}) \\ &= \sum_{d=0}^{s-1} \phi_{s-1d}(d(a^+_d + a^0_d + a^-_d) + (a^+_d - a^-_d)) \\ &= \sum_{d=0}^{s-1} \phi_{s-1d}(d + (a^+_d - a^-_d)) \end{aligned} \tag{9}$$

And analogously:

$$\begin{aligned} \Delta_2(s) &= \sum_{d=0}^{s} d^2(a^+_{d-1}\phi_{s-1d-1} + a^0_d\phi_{s-1d} + a^-_{d+1}\phi_{s-1d+1}) \\ &= \sum_{d=0}^{s-1} \phi_{s-1d}(d^2 + 2d(a^+_d - a^-_d) + a^+_d + a^-_d) \end{aligned} \tag{10}$$

Filling in the explicit expressions for the coefficients (see Eq. 7) we obtain:

$$\Delta_1(s) = (1 - \frac{1}{n})\Delta_1(s-1) + (1 - \frac{1}{P}) \tag{11}$$

$$\Delta_2(s) = (1 - \frac{2}{n})\Delta_2(s-1) + (2 - \frac{2}{P} + \frac{2}{nP} - \frac{1}{n})\Delta_1(s-1) + (1 - \frac{1}{P}) \tag{12}$$

The fixed points of these difference (or recursion) equations are unique and correspond to the limit $s \to \infty$, or equivalently, random sampling, They are found to be

$$\Delta_1(\infty) =< d(A,B) >_{random} = n(1 - \frac{1}{P}) \tag{13}$$

$$\Delta_2(\infty) =< d^2(A,B) >_{random} = \frac{n(P-1)(1-n+nP)}{P^2} \tag{14}$$

where A and B are random configurations (with $d(A,B)$ the distance between A and B).
We can define the corresponding relaxation functions $q_k(s)$ [19] as follows:

$$q_k(s) = \frac{< d^k(A,B) >_{random} - < d^k(A_0,A_s) >}{< d^k(A,B) >_{random}} = 1 - \frac{\Delta_k(s)}{\Delta_k(\infty)} \tag{15}$$

where $k = 1, 2$ and A_0 and A_s are the initial and final point of a random walk of length s.
After rewriting, we arrive at the following recursion relations for the $q_k(s)$:

$$q_1(s) = (1 - \frac{1}{n})q_1(s-1) \tag{16}$$

$$q_2(s) = \frac{q_1(s-1)(2-2n+2nP-P)}{n-n^2(1-P)} + q_2(s-1)(1-\frac{2}{n}) \tag{17}$$

Clearly for $q_1(s)$ we can obtain immediately:

$$q_1(s) = (1 - \frac{1}{n})^s = e^{-s/\tau_1} \tag{18}$$

Where $\tau_1 = \frac{1}{ln(n/n-1))} \approx n$.
To arrive at a closed formula for $q_2(s)$ first the recursion relation for $q_2(s)$ is rewritten by the relation for $q_1(s)$:

$$q_2(s) = q_2(s-1)g + a^{s-1}b \tag{19}$$

where $b = \frac{(2-2n+2nP-P)}{n-n^2(1-P)}$, $a = (1 - \frac{1}{n})$ and $g = (1 - \frac{2}{n})$.
since $q_2(0) = 1$, we can derive that:

$$q_2(s) = g^s + ba^{s-1}\sum_{i=0}^{s-1}(\frac{g}{a})^i \tag{20}$$

In the sum term we recognise the geometrical series:

$$\sum_{i=0}^{s-1} x^i = \frac{1-x^s}{1-x} \tag{21}$$

which leads to the general expression:

$$q_2(s) = g^s(1 + \frac{b}{g-a}) - a^s\frac{b}{g-a} \tag{22}$$

which can be rewritten using two different relaxation times (τ_1 and τ_2).

$$q_2(s) = (1 + \frac{b}{g-a})e^{-s/\tau_2} - \frac{b}{g-a}e^{-s/\tau_1} \tag{23}$$

obviously,

$$\tau_1 = -\frac{1}{lna} \approx n \tag{24}$$

and

$$\tau_2 = -\frac{1}{lng} \approx \frac{n}{2} \tag{25}$$

3.2 Random walks through the TAP landscape

In the previous subsection we have restricted our attention to the correlation structure of distance sequences on the Hamming graph. In this section to every vertex in the graph a cost value will be assigned according to function H, e.g. Eq. 2.
Weinberger [21] proposed the autocorrelation function:

$$\rho(d) = \frac{< (H(A) - < H >)(H(B) - < H >) >_{d(A,B)=d}}{\sigma^2} \tag{26}$$

(where d is the number of random walks steps and σ^2 is the variance of H), as the most useful characteristic of a fitness landscape $H : C \rightarrow I\!R$. Apart from totally uncorrelated landscapes, $\rho(d) = \delta(d, 0)$, the simplest class consists of the nearly fractal AR(1) landscapes. A time series which is isotropic, Gaussian and Markovian will lead to an autocorrelation function of the form characterised by [21]:

$$\rho(d) \approx \rho(1)^d = e^{-d/\lambda}, d \ll n \tag{27}$$

where λ is the *correlation length*.
The definition of the autocorrelation function, (26), can be rewritten as

$$\rho(d) = 1 - \frac{< (H(A) - H(B))^2 >_{d(A,B)=d}}{2\sigma^2} \tag{28}$$

According to Eq. 27, the auto-correlation function for an AR(1) landscape can be determined from analysis of the 1-step auto-correlation. Let t and t' be two configurations with $d(t, t') = 1$ and corresponding costs H and H'. According to (28) we can write:

$$\rho(1) = 1 - \frac{< (H - H')^2 >}{2\sigma^2} = 1 - \xi \tag{29}$$

We assume that $\xi \ll 1$, which is reasonable, since we look at a small variation in H.
If ξ is sufficiently small we have

$$\lambda = -\frac{1}{ln\rho(1)} = -\frac{1}{ln(1-\xi)} \approx \frac{1}{\xi} \tag{30}$$

or equivalently,

$$\lambda = \frac{2\sigma^2}{< (H - H')^2 >} \tag{31}$$

3.2.1 The one step correlation function for the Task Allocation Problem

As previously stated, we consider the task allocation problem for a processor topology that is fully connected and homogeneous, so processor- and link speeds are set to unity. Furthermore the work per task is considered to be unity. We consider a general class of random task graphs. Each pair of tasks is connected with probability γ. In graph theoretical terms we consider simple graphs, so maximally one edge connects two vertices (tasks) and a task is not connected to itself.
The TAP phase space properties are studied using the cost function (2). If we mutate the allocation number of task k (in an arbitrary initial configuration) we can derive the following formula for the change in cost $\delta H = H - H'$:

$$\delta H = 2w_k(W_m - W_n - w_k) + 2R \tag{32}$$

if task k gets assigned a new allocation number. Else δH is 0.
w_k is the work associated with task k, m is the previous allocation number, n the new one, W_m is the execution time due to the work on processor m and equivalently for processor n. Both calculation time values are taken before the mutation. The term R denotes the change in the communication cost (communication cost before - communication cost after).
However, we are interested in $< (\delta H)^2) >$. After some algebra we obtain: (including the fact that only a fraction $(P-1)/P$ of the mutations contributes indeed the amount (32)).

$$< (\delta H)^2) >= \frac{P-1}{P}(4(1-2 < R > + < R^2 > +2(< W_n^2 > - < W_m W_n > + < W_n R > - < W_m R >))) \tag{33}$$

So, in order to obtain an analytical expression for (33) we need to calculate six quantities: $< R >$, $< R^2 >$, $< W_n^2 >$, $< W_m W_n >$, $< W_m R >$ and $< W_n R >$.
Before continuing with our derivation of the one-step auto-correlation, first an expression for σ^2 will be derived.
We have

$$\sigma^2 =< H^2 > - < H >^2 \tag{34}$$

The simplest of the two terms is $< H >^2$. We can see that

$$< H >= \sum_p < W_p^2 > + \sum_{p,q} < C_{pq} > \tag{35}$$

The probability that a given task i gets assigned a specific allocation number j is denoted by q, consequently the probability that the task doesn't get the allocation number is equal to $1-q$. So we can consider this as a binomial distribution.
The probability that k tasks get assigned to a specific processor number is therefore given by:

$$\left(\begin{array}{c} n \\ k \end{array} \right) q^k (1-q)^{n-k} \tag{36}$$

Obviously $q = \frac{1}{P}$. The expectation value for k is given by $< k >= nq = n/P$, whereas the variance $< k^2 > - < k >^2$ of k is equal to $\frac{n}{P}(1-\frac{1}{P})$.
This leads us directly to the following expression for $< k^2 >$:

$$< k^2 >= \frac{n}{P}(\frac{n}{P} + 1 - \frac{1}{P}) \tag{37}$$

which is equal to $< W_p^2 >$ in the case that all tasks have unit weight.
Next, consider $< C_{pq} >$. We are interested in the probability of having l tasks on some processor p, and k tasks on another processor q, sharing x edges. We denote by $P(x \cap (l \cap k))$ the probability that the above event occurs. We can express this probability as a product of two other probabilities using Bayes theorem:

$$P(x|l \cap k) = \frac{P(x \cap (l \cap k))}{P(l \cap k)} \tag{38}$$

So, the probability that we look for is $P(x \cap (l \cap k)) = P(x|l \cap k)P(l \cap k)$; the product of the probability that we have l nodes on some processor p and k nodes on some processor q, times the probability that given the first restriction the tasks on these processors share x edges. This leads to the following expression for the expected communication between an arbitrary processor pair:

$$< C_{pq} >= \sum_l \left(\begin{array}{c} n \\ l \end{array} \right) q_1^l (1-q_1)^{n-l} \sum_k \left(\begin{array}{c} n-l \\ k \end{array} \right) q_2^k (1-q_2)^{n-l-k} \sum_x \left(\begin{array}{c} lk \\ x \end{array} \right) \gamma^x (1-\gamma)^{lk-x} x \tag{39}$$

Where, $q_1 = \frac{1}{P}$ and $q_2 = \frac{1}{P-1}$ which reduces to

$$< C_{pq} >=< x >= \sum_l \left(\begin{array}{c} n \\ l \end{array} \right) q_1^l (1-q_1)^{n-l} \sum_k \left(\begin{array}{c} n-l \\ k \end{array} \right) q_2^k (1-q_2)^{n-l-k} \gamma k l \tag{40}$$

And therefore

$$< C_{pq} >= \sum_l \left(\begin{array}{c} n \\ l \end{array} \right) q_1^l (1-q_1)^{n-l} \gamma l (n-l) q_2 \tag{41}$$

Simplifying to

$$< C_{pq} >= \gamma n q_2 < l > -\gamma q_2 < l^2 > \tag{42}$$

We already saw that $< l^2 >= \frac{n}{P}(\frac{n}{P} + 1 - \frac{1}{P})$ and $< l >= \frac{n}{P}$, so

$$< C_{pq} >= \gamma \frac{n(n-1)}{P^2} \tag{43}$$

This gives us the following expression for $< H >$, where we take into account that the $< W_p^2 >$ term counts P times, and the $< C_{pq} >$ term counts $P(P-1)$ times.

$$< H >= n(\frac{n}{P} + 1 - \frac{1}{P}) + \gamma \frac{(P-1)n(n-1)}{P}) \tag{44}$$

Next an expression for $< H^2 >$ will be derived.

$$< H^2 >=< \sum_{p,q} W_p^2 W_q^2 + 2 \sum_{p,o,m} W_p^2 C_{mo} + \sum_{q,r,m,o} C_{qr} C_{mo} > \tag{45}$$

or,

$$< H^2 >=< \sum_{p,q} W_p^2 W_q^2 > +2 < \sum_{p,o,m} W_p^2 C_{mo} > + < \sum_{q,r,m,o} C_{qr} C_{mo} > \tag{46}$$

The first term can be rewritten in two separate sums. We must distinguish the possibilities $p = q$ and $p \neq q$.

$$\sum_{p,q} W_p^2 W_q^2 = \sum_p W_p^4 + \sum_{p \neq q} W_p^2 W_q^2 \tag{47}$$

Let's consider the case of $p = q$. Assuming k tasks on processor p we have

$$< W_p^4 >=< k^4 >= \sum_k \left(\begin{array}{c} n \\ k \end{array} \right) q^k (1-q)(n-k) k^4 \tag{48}$$

For a binomial distribution the kurtosis (4th moment)

$$< (k - < k >)^4 >= (nq(1-q))^2 (3 + \frac{1 - 6q(1-q)}{nq(1-q)}) = m_4 \tag{49}$$

And thus,

$$< k^4 >= m_4 + 4 < k^3 >< k > -6 < k^2 >< k >^2 + 3 < k >^4 \tag{50}$$

Furthermore the skewness (3rd moment) is given by

$$< (k - < k >)^3 >= nq(1-q)(1-2q) =< k^3 > -3 < k >< k^2 > +2 < k >^3 = m_3 \tag{51}$$

or,

$$< k^3 >= 3 < k >< k^2 > -2 < k >^3 + m_3 \tag{52}$$

Finally we find, since $< k >= nq$ and $< k^2 >= nq(nq + 1 - q)$ that

$$< W_p^4 >=< k^4 >= \frac{n\ \left(-6+11\,n-6\,n^2+n^3+12\,P-18\,n\,P+6\,n^2\,P-7\,P^2+7\,n\,P^2+P^3\right)}{P^4} \quad (53)$$

Next, consider $p \neq q$, that is $< W_p^2 W_q^2 >=< k^2 l^2 >$.
In an analogous manner one arrives at:

$$< W_p^2 W_q^2 >= \frac{(-1+n)\ n\ \left(6-5\,n+n^2-4\,P+2\,n\,P+P^2\right)}{P^4} \quad (54)$$

In case of the interference term $< W_p^2 C_{qr} >$, we must consider the cases $p \neq q \neq r$ and $p = q \neq r$.
For the first case we get:

$$< W_p^2 C_{qr} >= \frac{\gamma\,n\ \left(2-3\,n+n^2\right)\ (-3+n+P)}{P^4} \quad (55)$$

And for the second case :

$$< W_q^2 C_{qr} >= \frac{\gamma\ (-1+n)\ n\ \left(6-5\,n+n^2-6\,P+3\,n\,P+P^2\right)}{P^4} \quad (56)$$

Finally, we are left with the $< C_{qr} C_{st} >$ terms.
For this case we can distinguish the following (contributing) cases:

1. $q \neq s \neq r \neq t$, leading to terms of the form $< C_{qr} C_{st} >$
2. $q = s \neq r \neq t$, leading to terms of the form $< C_{qr} C_{qt} >$
3. $q = s \neq r = t$, leading to terms of the form $< C_{qr} C_{qr} >$

Analogous to the method above the following expressions can be derived.

$$< C_{qr} C_{qr} >= \frac{\gamma\ (-1+n)\ n\ \left(6\,\gamma-5\,\gamma\,n+\gamma\,n^2-4\,\gamma\,P+2\,\gamma\,n\,P+P^2\right)}{P^4} \quad (57)$$

$$< C_{qr} C_{qt} >= \frac{\gamma^2\,n\ \left(2-3\,n+n^2\right)\ (-3+n+P)}{P^4} \quad (58)$$

$$< C_{qr} C_{st} >= \frac{\gamma^2\,n\ \left(-6+11\,n-6\,n^2+n^3\right)}{P^4} \quad (59)$$

Having available all the essential terms for $< H^2 >$, we can now write down the full formula, taking into account the proper pre-factors:

$$\begin{aligned} < H^2 >= P < W_p^2 > + \\ P(P-1) < W_p^2 W_q^2 > + \\ 2(P(P-1)(P-2) < W_p^2 C_{qr} > + \\ 2P(P-1) < W_q^2 C_{qr} >) + \\ 2P(P-1) < C_{qr} C_{qr} > + \\ 4P(P-1)(P-2) < C_{qr} C_{qt} > + \\ P(P-1)(P-2)(P-3) < C_{qr} C_{st} > \end{aligned} \quad (60)$$

Filling out all terms and simplifying the expression we finally end up with the following expression for the variance σ^2:

$$< H^2 > - < H >^2= \frac{2\ (\gamma-1)\ (n-1)\ n\ (1+\gamma\ (P-1))\ (P-1)}{P^2} \quad (61)$$

Note that, because of the appearance of γ^2 terms, Eq. 61 can only be used to predict the variance of an ensemble of random graphs with fixed γ. This is due to the following fact

$$(\sum_i deg(i))^2 \neq \sum_i (deg(i))^2 \tag{62}$$

which states that the squared sum over the individual vertex degrees is generally not equal to the sum over the squared vertex degrees.
So in order to experimentally verify this result we must calculate the variance over multiple graph instances. The γ^2 term is not present in the expression for the average cost (Eq. 44), which implies that it is valid for a specific random graph instance.
Then let's turn to $< (\delta H)^2) >$. We can express this as follows:

$$< (\delta H)^2) >= 4\frac{(P-1)}{P}(< R^2 > -4 < lR > +2(< l^2 > - < kl >)) \tag{63}$$

In the averaging procedure, we consider δH for only those cases that one processor has $(l+1)$ tasks (so at least 1), and the processor that the transfer is to has k tasks.
The following expressions for the individual terms can be derived:

$$\begin{aligned} < R^2 >&= \frac{2\,\gamma\,(-1+n)}{P} \\ < lR >&= \frac{\gamma\,(-1+n)}{P} \\ < l^2 >&= \frac{(-1+n)\,(-2+n+P)}{P^2} \\ < kl >&= \frac{(-2+n)\,(-1+n)}{P^2} \end{aligned} \tag{64}$$

which leads to

$$< (\delta H)^2) >= \frac{8\,(-1+\gamma)\,(1-n)\,(-1+P)}{P^2} \tag{65}$$

And thus our one-step auto correlation:

$$\rho(1) = 1 - \frac{< (H-H')^2 >}{2\sigma^2} = 1 + \frac{2}{n\,(-1+\gamma-\gamma\,P)} \tag{66}$$

Applying Eq.(31) we find directly

$$\lambda = \frac{n}{2}(1+\gamma(P-1)) \tag{67}$$

We see that for fixed γ and P, λ is linearly proportional to the number of tasks n. Note that we have assumed that $P > 1$ in our derivation, otherwise $\rho(1)$ is not defined.
It is very important to observe that there are no dependencies of γ^2 in Eq. 65, which implies that the variance in γ (due to σ^2) does not get eliminated. Strictly speaking this means that the derived formula for λ does not correctly predict the correlation structure of the landscape for single task graph instances. However, the $n/2$ term is obviously present in Eq. 67, which corresponds to the correlation time τ_2 derived in section 3.1. In section 6 we shall see that this is also the correlation length found experimentally.

4 Physics of Task Allocation

It can be shown that the Hamiltonian (energy function) of a spin glass is similar to the cost function of a well known NP-complete problem: graph bi-partitioning [13]. The cost function of the graph bi-partitioning problem can be considered as a special instance of that of the TAP. In analogy with spin glasses and graph bi-partitioning the TAP Hamiltonian will be formulated.
Application and machine specific parameters are used to distinguish two different phases (a sequential- and a parallel allocation phase) in the spectrum of optimal task allocations. The location of the separation between the two phases as a function of the aforementioned parameters is determined by a mean field argument. This location gives a rough estimate of the transition region.
Many search methods have been shown to behave anomalytically for certain critical parameters of the instance of combinatorial search problems [22] (critical slowing down). We speculate on the existence of such an anomaly (often observable as a sudden increase in the search cost) in the spectrum of TAP instances.

4.1 Spin Glasses and Graph bi-partitioning

In the area of condensed matter physics, a canonical model to describe the properties of a magnet is the Ising model. In d dimensions this is a regular square lattice of atomic magnets, which may have spin up or spin down. Formally, we have n variables s_i, one for each individual magnet, where s_i can take on values $+1$ or -1. The Hamiltonian describing the magnetic energy present in a specific configuration, without an external magnetic field, is given by:

$$H = -\sum_{k>i} J_{ik} s_i s_k. \tag{68}$$

For the Ising spin model, the interaction strength J_{ik}, is constant. However, if the J_{ik}'s are independent negative and non- negative random variables, we obtain the spin glass Hamiltonian. The spin glass model exhibits frustration, opposed to the (square-lattice) Ising model. This specific characteristic of the Ising system causes only two ground states to be present in the Ising model (all spins up, or all spins down) and many (highly degenerate) ground states for the spin glass. While in the Ising model, each pair of aligned spins contributes the same amount of energy, this is not true for a spin glass. Alignment with one neighbouring spin can result in an energetically unfavourable situation with another neighbour.
A well known NP-complete problem, graph bi-partitioning, has a cost function which is equivalent to the Hamiltonian of the spin glass model. We consider a graph, a set of n vertices and E edges. A configuration is an equal partition of the vertices. This can be expressed with the following constraint:

$$\sum_i s_i = 0, \tag{69}$$

where $s_i = 1$ if vertex i is in partition 0 and $s_i = -1$ otherwise. The edges can be encoded with a connectivity matrix J_{ik}. Such that $J_{ik} = 1$ if i and k are connected and $J_{ik} = 0$ if not. The Hamiltonian of a configuration can be expressed as follows:

$$H = \sum_{i<k} J_{ik}(1 - s_i s_k)/2. \tag{70}$$

Eq. 70 is equal to the spin glass Hamiltonian (Eq. 68), up to a constant value of $\sum_{i<k} J_{ik}/2$. The constraint (69) introduces frustration, otherwise the cost would be minimal for all vertices in one partition. In other words, without the constraint we would have a simple Ising ferro-magnet.
For a detailed review of spin glass theory and graph bi-partitioning we refer to the book by Mezard *et al.* [13].

4.2 Task Allocation Hamiltonian

In analogy with the models above we can rewrite the task allocation cost function (2) as follows:

$$H = (1-\beta)\sum_{i>k} J_{ik}(1-\delta(s_i,s_k)) + \beta\sum_i W_i^2 \tag{71}$$

$$\delta(s_i,s_j) = \begin{cases} 1 & \text{if} \quad s_i = s_j \\ 0 & \text{otherwise,} \end{cases} \tag{72}$$

where $s_i \in \{1 \ldots P\}$, J_{ik} is the communication time between processors i and k and W_i the total calculation time on processor i.
Note that we have introduced a parameter β into the Hamiltonian. This β-parameter can be varied in the range $[0,1]$, in order to tune the amount of "frustration" between the calculation and the communication terms. Variations of β can be interpreted either as variation in an application's calculation-communication ratio or a machine's processor speed - bandwidth ratio [4].
The connection probability γ in a random graph, can be considered as a dual parameter for β. γ can be increased in the range $[0,1]$, which is equivalent to augmenting the average communication load, which can also be realised by decreasing β.

4.3 The TAP Phase Transition

Although the task allocation problem is NP-hard [1], the two extremes, $\beta = 0$ and $\beta = 1$ are easy to solve. For $\beta = 0$, the only relevant term in the Hamiltonian is an attracting communication term, which will cause all *connected* tasks to be allocated to one processor. For this extreme, the number of optima is exactly P. The corresponding lowest energy state will have value zero. This situation corresponds to a parallel machine with infinitely fast processors.
For $\beta = 1$ there is only a repulsive workload term, which will force that the variance in the workload distribution is minimised. This results in an equal partitioning of the total workload over all available processors. It can easily be shown that the total number of optima in this case equals:

$$\prod_{k=1}^{P} \begin{pmatrix} n \\ k(n/P) \end{pmatrix} = \frac{n!}{(n/P)!^P}. \tag{73}$$

It is assumed that the n tasks have unit weight and that n/P is integer. The corresponding optimal cost value obviously will be n^2/P.
In the case of $\beta = 0$ the P optima are maximally distant in terms of the defined distance metric (see section 3). The P-ary inversion operation (analogous to spin-flipping in spin glass theory) and arbitrary permutations applied to a given optimal configuration leave the value of the Hamiltonian invariant. Note that, in this case, the TAP landscape is highly symmetrical. The entire landscape consists of P identical sub-landscapes. Each sub-landscape has only one optimum, which is automatically the global optimum.
In case of $\beta = 1$, the optima are relatively close to one another. Again, we can distinguish two types of operations that leave the value of the Hamiltonian invariant. The first type is trivial, that is, permutation of tasks allocated to the same processor, since this corresponds to the same point in phase space. The second type may change the point in phase space. Examples of such operations are rotation of the sequence and permutation of two arbitrary tasks.
From the perspective of parallel computing it is most ideal when all processors are engaged in a computation. However, the employment of all available processors does not always correspond to the optimal solution due to the communication overhead. Both machine and application specific parameters, which can be summarised as the ratio between the communication and calculation time, determine this optimal value.
We can observe a transition from sequential to parallel allocation when β is increased from 0 to 1 (or equivalently, if γ is decreased from 1 to 0). In order to quantify this transition we have to define an *order parameter*, which is a measure for the *degree of parallelism* present in an optimal allocation.

We assume that all tasks and connection weights are unity and define the order parameter $\mathcal{P}$, quantifying the *parallelism* in a given optimal allocation:

$$\mathcal{P} = 1 - \frac{(< W^2 > - < W >^2)P^2}{n^2(P-1)}. \tag{74}$$

where W is the time spent in calculation and $n^2(P-1)/P^2$ is the maximal possible variance in W. Eq. 74 takes the value 1 in the case of optimal parallelism ($\beta = 1$ or $\gamma = 0$) and the value 0 ($\beta = 0$ or $\gamma = 1$) in the case of a sequential allocation.

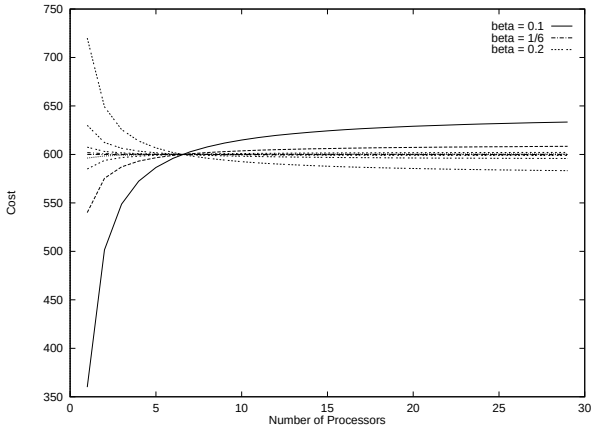

Figure 1: $< H >$ vs. P for increasing β, $n = 60$ and $\gamma = 0.2$

Using Eq. 75 which expresses the average cost, which was derived in section 3, we can calculate whether the average cost will either increase or decrease by using more processors for an allocation. Note that β has been included into Eq. 44. We expect that the transition from sequential to parallel allocation will approximately occur for those values of β and γ for which the average cost will change from a monotonically decreasing function to a monotonically increasing function of P.

$$< H >= \beta n(\frac{n}{P} + 1 - \frac{1}{P}) + (1-\beta)\gamma\frac{(P-1)n(n-1)}{P}. \tag{75}$$

We use Eq. 75 to predict for which values of γ and β the transition will occur approximately. In Fig. 1 an example of this transition is depicted, for a task graph with $\gamma = 0.2$, $n = 60$. The transition point as predicted will approximately occur for the following values of β and γ keeping one of the two variables fixed with the additional constraint that $\frac{\partial < H >}{\partial P} = 0$.

$$\beta_c = \frac{\gamma}{1+\gamma} \tag{76}$$

$$\gamma_c = \frac{\beta}{1-\beta} \tag{77}$$

We interpret β_c and γ_c as the "critical" values of β and γ in analogy with e.g. the critical temperature T_c in thermal phase transitions or percolation threshold p_c in percolation problems. Note that in Fig. 1 there is a point where the average value of the Hamiltonian is independent of β (approximately at $P = 7$). This is due to the fact that Eq. 75 contains β independent terms and therefore the β dependent terms can be eliminated for certain values of P, given fixed n and γ.

5 Experimental Methods

In this section several experimental methods that will be used in our study are introduced. Firstly, SA which is used to find sub-optimal solutions to the TAP. Secondly, a method is presented to quan-

tify the computational search cost. Thirdly, we will briefly discuss an experimental method to determine the correlation length of the phase space of a TAP instance.

5.1 Simulated Annealing For Optima Search

In simulated physical systems, configurations at thermal equilibrium can be sampled using the Metropolis algorithm [12]. The determination of the location of the critical temperature, can be established by sampling at fixed temperatures over some temperature range.
In the case of task allocation we are not interested in finding equilibria, but optimal configurations. For this purpose, exhaustive search is the only correct method that can be used for finding global optima. Unfortunately, this can be very inefficient and in worst case requires exponentially large search times. Therefore another more effective search method has to be selected.
In previous work [3][15], we have applied a Parallel Cellular Genetic Algorithm (PCGA) to find optimal solutions to the TAP in a parallel finite element application. Another possibility is using SA. The usefulness of SA on the TAP depends on the shape of the phase space. In section 3 we argued that the landscape has a self-similar structure, which is an indication of good performance of local heuristic search techniques. Indeed we have found that SA, was superior to GA, both in efficiency and quality of the solution. Therefore SA is applied to find the (sub) optimal solutions.

5.2 Search Cost Estimation

In comparable (NP-hard) problems the computational cost of determining the (optimal) solutions shows a dependence on problem specific parameters [23][6][2]. For example, in the case of graph colouring it has been observed that the "difficulty" of determining if a graph can be be coloured, increases abruptly when the average connectivity in the graph is gradually increased to some critical value [22].
Another example of a system where computational cost is affected by such parameters is that of a physical system where a thermal phase transition occurs (like the Ising model). The difficulty of finding the equilibrium value increases when the critical point is approached, and theoretically (in the thermodynamic limit) will become infinite at the critical point. This is generally referred to as critical slowing down.

In analogy with this behaviour we expect that in the task allocation problem comparable phenomena can be found in a critical region of the β and γ-domain. For both β extremes the optima are known in advance. The difficulty to find these optima is therefore reduced to order unity. If the calculation and the communication term in the Hamiltonian (71) are of comparable magnitude we can say that the system is in a critical (or frustrated) area. Moving away from this critical region one term becomes small noise for the other.
We are interested in a method for obtaining an estimate of the computational cost (difficulty) of finding optima for given problem parameters. In order to quantify the search cost, we measure the number of local optima, in which independent steepest descent runs get stuck. A specific search space is considered to be "simple" if it contains a relatively small number of local optima. On the other hand, if the number of local optima is large the corresponding search space is classified as "difficult". The distinction between local optima is based on the cost of the corresponding task allocations. That is two allocations i and j (that are local optima) are called distinct if:

$$H(i) \neq H(j) \tag{78}$$

In the experiments below, the number of steepest descent runs is taken to be $10n$, with n the number of tasks.

5.3 Measuring Phase Space Structure

The structure of the TAP phase space is characterized using the auto-correlation function (79) of a random walk.

$$\rho(d) = \frac{< H(A)H(B) >_{d(A,B)=d} - < H >^2}{\sigma^2}, \tag{79}$$

where $d(A, B)$ is the "distance" between two configurations A and B as introduced in section 3. The value for λ for the task allocation phase space can be directly determined from $\rho(1)$.

6 Experimental Results

In this section experimental results regarding the statistical quantities, correlation length, phase transition and search cost for the TAP are presented.
First a number of experiments, conducted to verify the analytical results derived in section 3 are given. It is established that the TAP landscape is AR(1), which supports the argument for using SA in the subsequent experiments for finding optimal allocations.
The occurrence of the phase transition for several parameter values is observed in the corresponding experiments. Complementary to the phase transition is the divergence of the computational cost, which is also shown to manifest itself.

6.1 Statistical Quantities and Correlation Length

In for example the *Travelling Salesman Problem* (TSP) [19] statistical quantities of the landscape of random TSP instances can be obtained by random walk averaging. This is not possible for TAP. Only for both connectivity extrema, $\gamma = 0.0$ and $\gamma = 1.0$, the random walk is self averaging, which means that the ensemble average can be obtained by a random walk. For other values of γ each instance of a random graph differs in connectivity from the other, which implies that statistical quantities can only be estimated by averaging over multiple instances of random graphs.

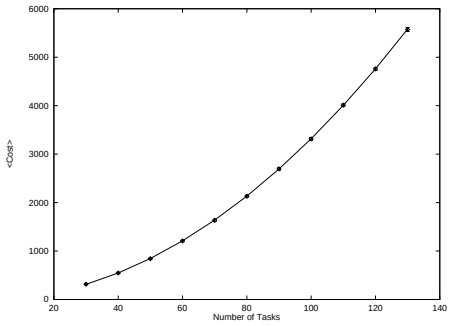

Figure 2: $< H >$ for different n

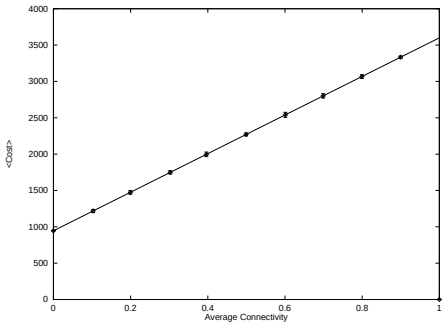

Figure 3: $< H >$ for different γ

The determination of the auto-correlation functions is obtained using a specific instance of the TAP with fixed γ, n and P. We can not use the derived formula for the variance to predict the variance of a single TAP instance, this is due to the presence of γ^2 terms in the expression for σ^2 (see Eq. 61). Such terms are not present in the formulae for $< H >$ (Eq. 44) and $< (\delta H^2) >$ (Eq. 65).
In all figures error bars are displayed, if applicable.

6.1.1 Experimental Verification of Cost Terms

In this section we experimentally verify the derived expressions (Eqs. 44 and 60) for the expected cost and expected squared cost. Furthermore the equation for $< (\delta H)^2 >$ (Eq. 65) is experimentally

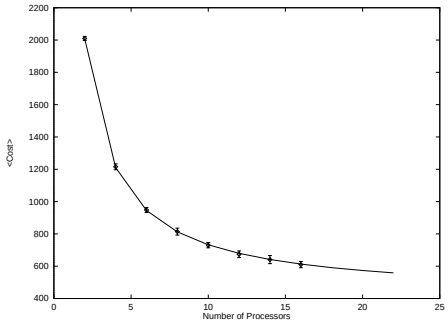

Figure 4: $< H >$ for different P

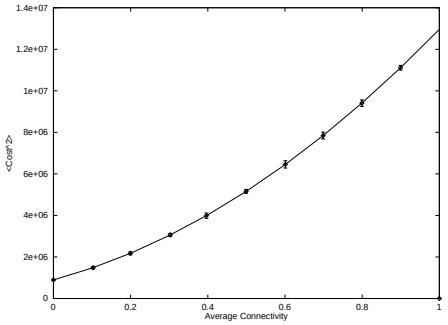

Figure 5: $< H^2 >$ for different γ

verified. We have carried out experiments with variable number of processors (P), connectivity (γ) and number of tasks (n). For each variable parameter the other two parameters were kept fixed ($n = 60$, $\gamma = 0.1$ and $P = 4$). The results are shown in Figs. 2-10.

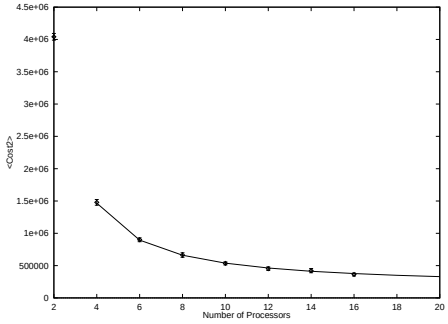

Figure 6: $< H^2 >$ for different P

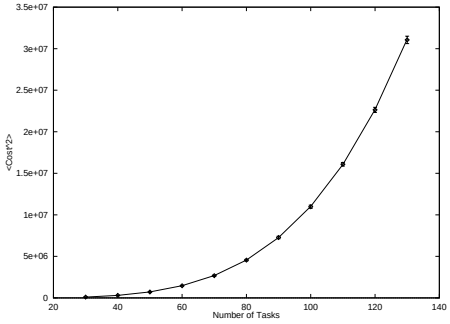

Figure 7: $< H^2 >$ for different n

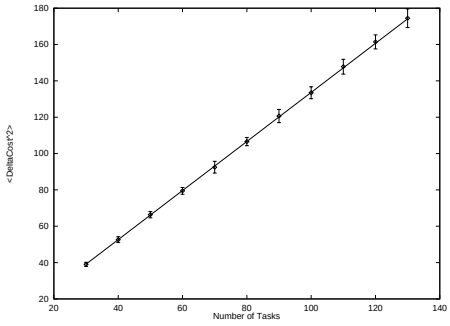

Figure 8: $< (\delta H)^2 >$ for different n

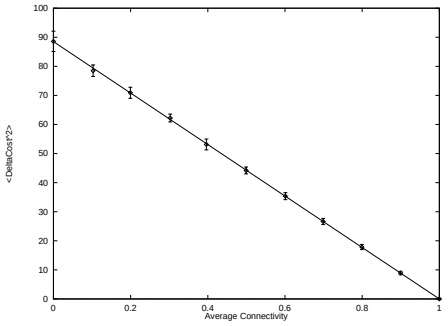

Figure 9: $< (\delta H)^2 >$ for different γ

6.1.2 Analytical and Measured λ

In this section the correlation length is experimentally determined. For these experiments random walks through the TAP landscape with approximate lengths of 10^5 steps were generated. Subsequently, the autocorrelation functions using the values encountered were calculated.
In Fig. 11 two measured and predicted correlation functions are displayed. In the first experiment we have used 100 tasks, 8 processors and a connection probability of 0. In the second experiment a TAP instance with a non-zero connection probability ($\gamma = 0.5$), $n = 64$ and $P = 4$ was used.

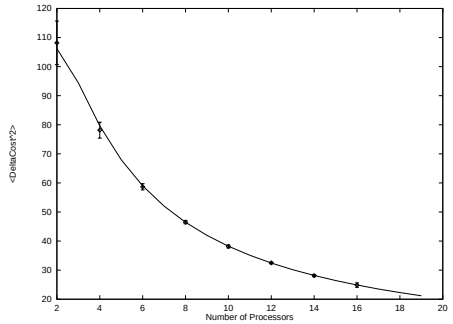

Figure 10: $< (\delta H)^2 >$ for different P

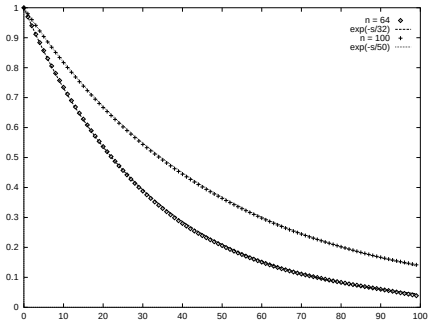

Figure 11: Analytical and experimental values for the autocorrelation function $n = 100$, $P = 8$ and $\gamma = 0.0$ and $n = 64$, $P = 4$ and $\gamma = 0.5$.

6.2 Phase Transitions and Computational Search Cost

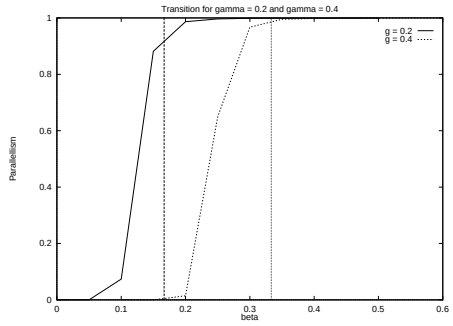

Figure 12: phase transition with fixed γ (0.2 and 0.4) and increasing β. The vertical solid lines indicates the location of the transition as predicted by Eq. 76

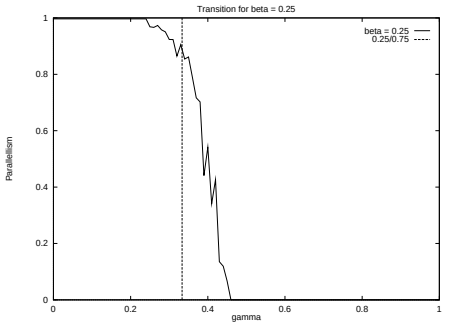

Figure 13: A phase transition with $\beta = 0.25$. The vertical solid line indicates the location of the transition as predicted by Eq. 76

Several experiments are performed to demonstrate the existence of a phase transition, and the location of the transition as predicted by Eq. 76 is checked. The experiments to which the depicted data corresponds were carried out with $n = 64$ and $P = 8$. In Fig. 12, β is varied in the range $[0, 1]$ and γ is fixed at two different values (0.2 and 0.4). In Fig. 13 the dual experiment is performed, now γ is varied in the range $[0, 1]$ and β is fixed at the value 0.25. The results presented are comparable with those found for arbitrary parameter values. The mean field transition points are plotted as vertical lines.
In Figs. 14 and 15 the divergence of the search cost near the transition point can be observed. The method described in section 5.2 is used to quantify the cost. In Fig. 14, $n = 32$ and $P = 4$ and γ is fixed to 0.5. An increase in the number of local optima is found around the location of the phase transition. Another example is shown in Fig. 15, where $n = 64$, $P = 8$ and β is fixed to 0.2. Again the computational cost increases in the neighbourhood of the phase transition.

7 Summary and Discussion

In analogy with graph bi-partitioning and spin-glass theory we have constructed a cost function that expresses task allocation quality into a Hamiltonian form. It has been argued that the TAP is

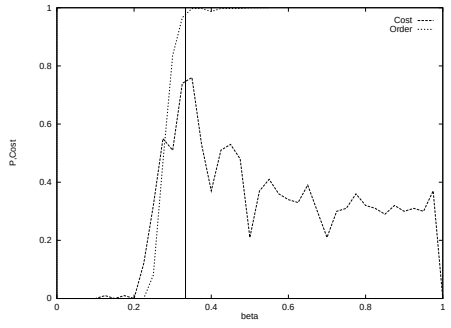

Figure 14: Computation cost diverges at the phase transition, $P = 4, n = 32, \gamma = 0.5$, β varied

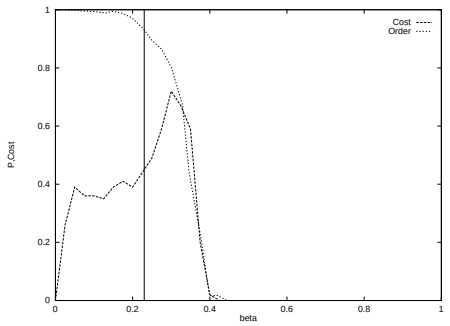

Figure 15: Another example with $n = 64$ and $P = 8$, $\beta = 0.2$, γ varied.

an example of so called frustrated systems, and as such is expected to show typical complex behaviour. The competition between the calculation and communication terms in the Hamiltonian is the source of frustration. In order to facilitate our study on frustration in the TAP a control parameter β was introduced into the Hamiltonian. The β parameter can be considered as a dual parameter for the degree of connectivity between tasks in the task graph. In the case of random task graphs γ is the connection probability between vertices (or tasks). The β parameter has an important interpretation in terms of high performance computing terminology. It either expresses an application's calculation-communication ratio or a machine's processor speed-bandwidth ratio.

In order to select a suitable method to find optima several aspects of the TAP phase space were investigated. Firstly, some basic characteristics like the size of the TAP phase space and its diameter were given. Secondly, the concept of AR(1) landscapes was discussed and the performance of SA on landscapes that exhibit such structure. We have derived the correlation length of a random walk trough the TAP phase space. First we derived analytical expressions for the relaxation functions on random walks through the TAP landscape. It was shown that the correlation length of the phase space corresponds to one of the two relaxation times (τ_2) found in this expression (Eq. 23). Secondly, a formal expression for both the variance of the cost and the squared difference in the cost between two subsequent allocations was derived.
The number of global optima for the extreme values of β in the Hamiltonian was discussed, as well as the invariance properties of the Hamiltonian in these cases. An order parameter $\mathcal{P}$, was introduced to quantify a degree of parallelism. Using an expression for the average value of the Hamiltonian (or cost) a rough measure was given for the location of the transition region where the optimal solution changes from sequential to parallel allocation.
Next, the observation was made that comparable systems show divergent behaviour in the computational cost that is associated with finding optimal values in a critical region, e.g. near a phase transition. It was argued that the transition of sequential to parallel allocation, induced by varying β or γ, is expected to give rise to analogous critical behaviour for the search cost in the TAP.

7.1 Statistical Quantities and Correlation Length

From Figs. 2-10 it is clear that the analytical formulae predict the corresponding average quantities to a high degree of accuracy. The specific choice of parameters does not influence the accuracy of the experiments. In other words, the specific problem instances for which the data are shown are indicative for the correctness of the derived expressions.
We only have an expression for the variance over an ensemble of random graphs with a fixed value of γ. This can not be used to predict the correlation length (λ) of the autocorrelation function for a random graph instance. Therefore, we can not derive an exact expression for the one step autocorrelation function. However the correlation time τ_2, found in Eq. 23 corresponds to the correlation

length that is found experimentally. Empirically, it can be shown that the variance of H over a random graph instance is approximately described by the following equation:

$$< H^2 > - < H >^2 = \frac{2\ (\gamma - 1)\ (n-1)\ n\ \ (P-1)}{P^2} \tag{80}$$

Eq. 80 correctly predicts the variance for a single random graph instance (data not shown). Using this equation (80) and Eq. 65 we do find the correct prediction of λ:

$$\rho(1) = 1 - \frac{2}{n} \rightarrow \lambda = \frac{n}{2} \tag{81}$$

The corresponding correlation length $\lambda = n/2$ indicates that the TAP landscape is smooth in comparison with that of graph bi-partitioning ($\lambda = n/4$). The correlation length is strongly related to the chosen perturbation mechanism. Increasing correlation length indicates a phase space that is smoother and easier to search. The fact that the search process benefits from larger correlation lengths, has been established in [20].

7.2 Phase Transitions and Computational Search Cost

As shown in Figs. 12 and 13, the approximate location of the phase transition that is induced by variation of β or γ can be predicted by a simple mean field argument (A practical consequence of this observation is, that given real application and machine parameters this can provide an estimate of the "usefulness" of parallel computation of the problem at hand).
In Figs. 14 and 15 it is shown that the presence of a phase transition is accompanied by the anomalous behaviour of the search cost. This behaviour is analogous to that observed in comparable combinatorial optimisation problems as argued above.

8 Concluding Remarks and Future Work

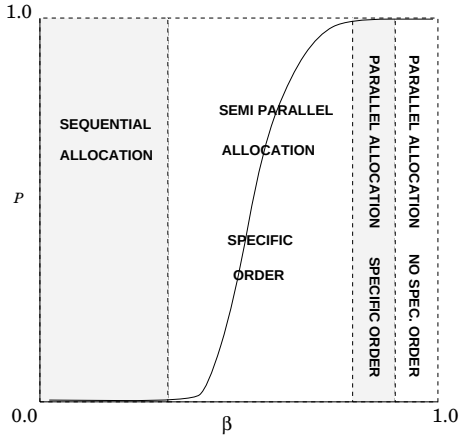

Figure 16: The different allocation regimes in the task allocation problem for varying β

It has been shown that the landscape of the TAP can be classified as an AR(1) landscape. Given the fact that the correlation length is $n/2$ for the defined cost function and neighbourhood structure, the TAP is an easier problem to solve using local search methods than e.g. graph bi-partitioning. In other work [19], it has been shown that both the landscape of spin-glasses and graph bi-partitioning have a correlation length of $n/4$, which makes those landscapes less smooth than that of the TAP.
The results presented in this paper clearly show that the task allocation problem exhibits a variety of interesting properties. For specific parameter sets the task allocation problem exists only in a small parameter range. Outside this range the problem is trivial. The problem becomes complex in the

range where the calculation and communication terms are of comparable magnitude. The location of this complex region is marked by the presence of a transition from sequential to parallel allocation. The different allocation regimes are summarised in Fig. 16. The *sequential allocation* region only contains optima where all tasks are allocated to one processor. The *semi-parallel allocation* region can correspond to the following situation. Not all available processors are necessarily used, due to the high competition with the communication cost. Also the locality in the task graph has its consequences for the allocation sequence. Tasks that are connected to one another "desire" to be grouped on the same processor. The last region, *parallel allocation*, corresponds to the mode where the inter task connectivity has become insignificant. This may either be due to a high speed communication network or weakly connected task graph. In this case optimal allocations can be realized using scattered decomposition.
In the near future we intend to investigate the effects of introducing locality in both the task graph as well as the processor topology. Also, we will study the critical behaviour in more detail, by means of finite size scaling experiments [9]. Furthermore, our interest goes out to a thorough understanding of task allocation in dynamic and heterogeneous parallel applications and machines, in e.g. cluster computing [14].

References

[1] D. Bernstein, M. Rodeh, and I. Gertner. On the complexity of scheduling problems for parallel/pipelined machines. *IEEE Transactions on Computers*, C-38(9):1308–1313, 1989.

[2] P. Cheeseman, B. Kanefsky, and W.M. Taylor. Computational complexity and phase transitions. In *Workshop on Physics and Computation*. IEEE Computer Society, 1992.

[3] J.F. de Ronde, A. Schoneveld, P.M.A. Sloot, N. Floros, and J. Reeve. Load balancing by redundant decomposition and mapping. In H. Liddell, A. Colbrook, B. Hertzberger, and P. Sloot, editors, *High Performance Computing and Networking*, volume 1067 of *Lecture Notes in Computer Science*, pages 555–561, 1996.

[4] G. Fox, M. Johnson, G. Lyzenga, S. Otto, J. Salmon, and D. Walker. *Solving Problems on Concurrent Processors*, volume 1. Prentice-Hall, 1988.

[5] R.W. Hamming. Error detecting and error correcting codes. *Bell Syst. Tech. J.*, 29:147–160, 1950.

[6] T. Hogg. Statistical mechanics of combinatorial search. In *Workshop on Physics and Computation*, Dallas, Texas, 1994.

[7] J. De Keyser and D. Roose. Load balancing data parallel programs on distributed memory computers. *Parallel Computing*, 19:1199–1219, 1993.

[8] S. Kirkpatrick, C.D. Gelatt jr., and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.

[9] W.G. Macready, A.G. Siapas, and S.A. Kauffman. Criticality and parallelism in combinatorial optimization. *Science*, 271:56–59, 1996.

[10] N. Mansour and G. Fox. A hybrid genetic algorithm for task allocation. In *Proceedings of the 4th International Conferencen on Genetic Algorthims*, pages 466–473, July 1991.

[11] N. Mansour and G. Fox. Allocating data to multicomputer nodes by physical optimization algorithms for loosely synchronous computations. *Concurrency: practice and experience*, 4(7):557–574, 1992.

[12] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journ. of Chemical Physics*, 21:1087–1092, 1953.

[13] M. Mezard, G. Parisi, and M.A. Virasoro. *Spin Glass Theory and Beyond*. World Scientific, 1987.

[14] B.J. Overeinder, P.M.A. Sloot, R.N. Heederik, and L.O. Hertzberger. A dynamic load balancing system for parallel cluster computing. *Future Generation Computer Systems*, 12(1):101–115, 1996.

[15] A. Schoneveld, J.F. de Ronde, P.M.A. Sloot, and J.A. Kaandorp. A parallel cellular genetic algorithm used in finite element simulation. Accepted for Parallel Problem Solving from Nature (PPSN96).

[16] D. Sherrington. Complexity due to disorder and frustration. In E. Jen, editor, *1989 Lectures in Complex Systems*, volume II of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 415–453. Addison-Wesley, 1990.

[17] P.M.A. Sloot, J.A. Kaandorp, and A. Schoneveld. Dynamic complex systems (dcs): A new approach to parallel computing in computational physics. Technical Report TR-CS-95-08, University of Amsterdam, 1995.

[18] G.B. Sorkin. Simulated annealing on fractals: Theoretical analysis and relevance for combinatorial optimisation. In W.J. Dally, editor, *Advanced Research in VLSI*, Proceedings of the 6th MIT Conference, pages 331–351, 1990.

[19] P.F. Stadler and R. Happel. Correlation structure of the landscape of the graph-bipartitioning problem. *J. Phys. A: Math. Gen.*, 25:3103–3110, 1992.

[20] P.F. Stadler and W. Schnabl. The landscape of the traveling salesman problem. *Physics Let. A*, 161:337–344, 1992.

[21] E.D. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biol. Cybern.*, 63:325, 1990.

[22] C.P. Williams and T. Hogg. Using deep structure to locate hard problems. In *Proc. of the 10th Nat. Conf. on Artificial Intelligence*, pages 472–477, San Jose, California, 1992.

[23] C.P. Williams and T. Hogg. Phase transitions and coarse-grained search. In *Workshop on Physics and Computation*, Dallas, Texas, 1994.